PhD-FSTM-2022-102
The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 02/09/2022 in Esch-sur-Alzette
to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Jim Jean-Pierre BARTHEL
Born on 5 April 1994 in Luxembourg, Luxembourg

TOPICS IN COMPUTATIONAL NUMBER THEORY AND CRYPTANALYSIS

ON SIMULTANEOUS CHINESE REMAINDERING, PRIMES, THE MINTRU ASSUMPTION, AND FUNCTIONAL ENCRYPTION

Dissertation defence committee:

Dr Franck LEPRÉVOST, Chairman
Professor, Université du Luxembourg

Dr Gabor WIESE, Vice-Chairman
Professor, Université du Luxembourg

Dr Volker MÜLLER, Dissertation Supervisor
Professor, Université du Luxembourg

Dr Castryck WOUTER, Member
Researcher, KU Leuven, Belgium

Dr Nico DÖTTLING, Member
Researcher, CISPA Helmholtz Center for Information Security, St. Ingbert, Germany

Affidavit

I, Jim Jean-Pierre Barthel, hereby confirm that the PhD thesis entitled "Topics in Computational Number Theory and Cryptanalysis – On Simultaneous Chinese Remaindering, Primes, The MiNTRU Assumption, and Functional Encryption" has been written independently and without any other sources than cited.

Luxembourg, Esch-sur-Alzette
September 2, 2022

Place and Date

Signature

Acknowledgements

This dissertation would not have been possible without my supervisor Volker Müller. Thank you, Volker. I am grateful for your continuous support throughout my doctoral studies, the many hours of inspiring discussions, and all the lessons you have taught me over the years. I greatly enjoyed our conversations on new projects, the freedom you gave me to explore new directions, and your patience with my academic progress. I would like to express my deepest thanks to my collaborators Marc Beunardeau, Volker Müller, Răzvan Roşie, and Rajeev Anand Sahu for reaching out to me with new ideas, for their tremendous help with our articles, and the effort for publications. It was a pleasure to work with you. I would like to extend my deepest gratitude to my non-supervising CET members Franck Leprévost and Gabor Wiese. Thank you, Franck, for your linguistic guidance and your extraordinary attention to detail. Thank you, Gabor, for raising my interest in number theory during my Bachelor's studies and for your constructive feedback during my PhD. I would also like to pay my special regards to my two other jury members Nico Döttling and Wouter Castryck. I am thankful for your interest in my work and your enthusiasm to become jury members. I'd also like to express my gratitude to all my colleagues, for their support and the great work environment. I am particularly indebted to Agnese, Aoran, Bruno, Daniel, Emiliano, Franck, François, Gabor, Guenda, Jean-Luc, Jean-Sébastien, Lorenzo, Luca, Marc, Massimo, Najmeh, Nicolas, Paul, Pierrick, Pietro, Rajeev, Raluca, Răzvan, Reynaldo, Sebastiano, Sevdenur, Vitor, and Volker. I greatly enjoyed your company during all those years and could not have found better friends. I am grateful for various joyful moments, fun activities, trips, games, lunches, and dinners.

I must also thank Luca Notarnicola and Vincent Koenig for their valuable support in coordinating events in the SP2 community. I'd also like to acknowledge the precious help of the DCS secretaries Andrea Puech, Fabienne Schmitz, and Isabelle Schroeder, the doctoral school secretary Catherine Violet, and the DESCOM project officer Nicole Paschek. I am very grateful to the doctoral school and the BED who reacted promptly to my requests and enabled a smooth journey throughout my doctorate.

I would like to thank anyone interested in this thesis or my other works. I will be happy to answer any questions, discuss comments, and enter into collaborations. Research would not be half as interesting without you.

At last, I would like to thank all my family members for their incredible support and for always believing in me. I express a particular thank you to my girlfriend Catherine who always cheered me up during bad times and shared my joy during good times, who tolerated my work rhythm and helped me to clear my mind when needed. Thank you for being there for me.

Jim Barthel

This work has been supported by the Luxembourg National Research Fund through the grant PRIDE15/10621687/SPsquared.

Preface

The content of this thesis lies in the intersection of mathematics, computer science, and cryptology. It combines mathematical formalization with practical methods to achieve new results in complexity theory, computational number theory, and cryptography. The thesis consists of four main parts, each reporting on a different work. Those works are completely independent and differ from multiple points of view such as their background, their objectives, and their results. Their presentation style changes according to the target audience. The main parts are complemented by an auxiliary part that prepares several results required in the thesis. The individual parts are related through a short story of Jay, a young, soon to be, numismatic illustrating the everyday use of scientific results.

Part I: Toolbox

Part I prepares a collection of results required in the main parts. Chapter 1 fixes the notations and conventions. Chapter 2 prepares some statistical concepts. Chapter 3 revises some fundamental complexity notions, and Chapter 4 introduces the so-called 3-SAT problem required in Part II. In Chapter 5, two particular mathematical results are developed. To be precise, Theorem 5.1 yields the number of solutions of a linear multivariate modular equation and Theorem 5.2 upper bounds the volume of a high-dimensional shpere. Chapter 6 introduces (integer row) lattices, outlines their most important properties, and develops some elementary results. Chapter 7 concentrates on q-ary lattices. Based on the results from Chapter 5, two unconditional probabilistic lower bounds for a shortest lattice vector $\mathbf{v}$ of a q-ary lattice $\Lambda_q(\mathbf{A})$ generated by a random matrix $\mathbf{A} \in \left(\mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]\right)^{k \times m}$ are obtained. Theorem 7.3 shows that $\|\mathbf{v}\|_\infty \geq \frac{q^{\frac{m-k}{m}}}{4}$ with probability at least $1-2^{-m}$ and Theorem 7.5 proves that $\|\mathbf{v}\|_2 \geq \min\left\{q, \sqrt{\frac{m}{8\pi e}}\, q^{\frac{m-k}{m}}\right\}$ with probability at least $1 - \frac{1}{\sqrt{\pi m}} 2^{-m}$. The latter result is required in Part IV.

Part II: Simultaneous Chinese Remaindering

Part II introduces and studies a new computational problem called the Simultaneous Chinese Remainder Problem. Chapter 8 starts by revising the traditional Chinese Remainder Problem which asks to retrieve, if it exists, the unique integer $x \in \mathcal{S}_M \subset \mathbb{Z}$ such that $x \equiv r_i \mod m_i$ for all $i \in \{1, \ldots, k\}$ where $m_1, \ldots, m_k$ denotes a given list of moduli, $r_i \in \{0, \ldots, m_i - 1\}$ represents a remainder modulo m_i, and $\mathcal{S}_M$ denotes a representative solution set consisting of $\operatorname{lcm}(m_1, \ldots, m_k)$ consecutive integers. In Chapter 9, this problem is generalized to the Simultaneous Chinese Remainder Problem by replacing for each $i \in \{1, \ldots, k\}$ the single remainder r_i by a nonempty set of remainders $\mathcal{R}_i \in \{0, \ldots, m_i - 1\}$ any remainder of which may be chosen to satisfy the congruence modulo m_i. Section 9.2 argues that the number of corresponding solutions in $\mathcal{S}_M$ grows exponentially in the size of the remainder sets, thus causing any solving method that computes all solutions to run in exponential time. In Chapter 10, two particular decision problems for the Simultaneous Chinese Remainder Problem are studied. The Existential Simultaneous Chinese Remainder Problem, described in Definition 10.1, asks to decide whether there exists a solution for a given Simultaneous Chinese Remainder Problem. As the moduli may not be coprime, its solution may not be trivial. On the contrary, Theorem 10.2 proves that this problem is $\mathsf{NP-complete}$. The Bounded Simultaneous Chinese Remainder Problem, described in Definition 10.7, restricts to coprime moduli for which solutions are guaranteed to exist and asks to decide whether there exists a solution of a given size. Theorem 10.8 shows that also this problem is $\mathsf{NP-complete}$. In Chapter 11, multiple Simultaneous Chinese Remainder search problems are considered. For simplicity, these problems are restricted to coprime moduli and choose the trivial representative solution set $\mathcal{S}_M = \{0, \ldots, (\prod_{i=1}^{k} m_i) - 1\}$. The Minimal Simultaneous Chinese Remainder Problem, introduced in Definition 11.1, asks to find the minimal solution of a given Simultaneous Chinese Remainder Problem. Corresponding problem variants asking to find the maximal solution or a solution inside a given interval are also formalized and are shown in Proposition 11.5 to be polynomially equivalent to the Minimal Simultaneous Chinese Remainder Problem. The Minimal Simultaneous Chinese Remainder Problem in turn is $\mathsf{NP-hard}$ by Theorem 11.2. Despite the obvious hardness of the general problem, subsequent chapters study the Minimal Simultaneous Chinese Remainder Problem in further detail. Chapter 12 outlines a rough upper bound for the minimal solution. Chapter 13 yields the proof-of-concept that the minimal solution can be found without the need of comparing all the solutions of a given Simultaneous Chinese Remainder Problem. Chapter 14 outlines an observation on mixed-radix comparison systems allowing to reformulate a given Minimal Simultaneous Chinese Remainder Problem into a function minimization problem. Indeed, Corollary 14.8 shows that a particular family of functions over the remainder sets is minimized with respect to the remainders of the minimal or maximal solution. In Chapter 15, two existing lattice-based solving methods are presented and one of them is improved. In Chapter 16, the Simultaneous Chinese Remainder Problem is reformulated as a sieving problem, and it is shown that the maximal solution can be represented as the solution of a particular multiple-choice subset sum problem. Chapter 17 contains an illustrative example that summarizes the previous development. In Chapter 18, some empirical data is visualized and some heuristics for the Simultaneous Chinese Remainder Problem are discussed. Finally, in Chapter 19, we discuss some open questions and future research directions.

Part III: A Conjecture On Primes In Arithmetic Progressions And Geometric Intervals

Part III introduces a new conjecture on prime numbers. In Chapter 20, a synopsis of the results on the distribution of primes is presented with a focus on primes in arithmetic progressions. Chapter 21 starts by recalling Linnik's theorem which claims that there are absolute constants C and L such that for any integer $q \geq 2$ and any integer $1 \leq a \leq q-1$ with $\gcd(a,q) = 1$, the smallest prime $p_0 \equiv a \mod q$ satisfies $p_0 \leq Cq^L$. Conjecture 21.2 generalizes this claim by postulating that for every integer $t \geq 2$ a prime of this form can be found in the interval $[q^t, q^{t+1})$. In Chapter 22, Conjecture 21.2 is strengthened through some partial proofs. Lemma 22.1 proves that it holds for every sufficiently large exponent t, Theorem 22.7 improves the required lower bound on t under the Extended Riemann Hypothesis, and Lemma 22.9 computationally verifies the conjecture for all $q < 45000$. Theorem 22.10 combines those partial results to conclude that the conjecture holds unconditionally for every modulus $q < 45000$ or every sufficiently large exponent t and Theorem 22.11 shows that it holds under the Extended Riemann Hypothesis for every $t \geq 3$, as well as, for $t = 2$ for sufficiently large q. In Chapter 23, the relation of Conjecture 21.2 to other classical conjectures is studied. Theorem 23.4 shows that if Conjecture 21.2 holds, then Pomerance's conjecture, postulating that $(p_k - 1)$ divides $\prod_{i=1}^{k-1} p_i(p_i - 1)$ for any $k \geq 2$, where p_i denotes the i-th prime, also holds. Furthermore, Corollary 23.5 concludes that Pomerance's conjecture holds for any prime p_k such that the largest square factor of $p_k - 1$ is 45000-smooth. Chapter 24 describes some prospective research on primes in arithmetic progressions.

Part IV: On The (M)iNTRU Assumption Over Finite Rings

Part IV devises an attack against two recent computational hardness assumptions called the inhomogeneous NTRU assumption and the matrix inhomogeneous NTRU assumption. Chapter 25 loosely situates those new hardness assumptions in the cryptographic field. Chapter 26 formalizes the assumptions, outlines some existing applications, and specifies the assumption of our study – the $\mathsf{(M)iNTRU}$ assumption – consisting in a particular case of the original assumptions. For $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$, the $\mathsf{(M)iNTRU}$ search problem formalized in Section 26.3 asks one to retrieve $s \in \mathbb{Z}_q^{\times}$ from a given vector $(a_0, \ldots, a_\ell) \in \mathbb{Z}_q^{\ell+1}$ where $a_0 := \left[s^{-1} e_0 \mod q\right]$ and $a_i := \left[s^{-1}(2^{i-1} - e_i) \mod q\right] \quad \forall i \in \{1, ..., \ell\}$ for some given values $e_0, \ldots, e_\ell \in \mathbb{Z}_q$ following a distribution χ that produces with overwhelming probability elements of small norm. The $\mathsf{(M)iNTRU}$ decision problem requests to distinguish such a vector from a uniformly at random chosen vector in $\mathbb{Z}_q^{\ell+1}$. In Chapter 27, a first lattice attack against the decision problem is devised. Given a challenge vector $\mathbf{x} := (x_0, \ldots, x_\ell)$, Section 27.1 first constructs another vector $\mathbf{y} := (y_0, \ldots, y_{\ell-1})$ by setting $y_0 := x_0$ and $y_i := [2x_i - x_{i+1} \mod q]$ for all $i \in \{1, \ldots, \ell - 1\}$, and then considers the q-ary lattice $\Lambda_q(\mathbf{y})$. Section 27.2 devises a particular lattice basis of $\Lambda_q(\mathbf{y})$. In Section 27.3, it is concluded that if $\mathbf{x}$ was a random vector, then the shortest lattice vector is with high probability larger than $\min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}}\,q^{\frac{\ell-1}{\ell}}\right\}$. On the other hand, if $\mathbf{x}$ was a $\mathsf{(M)iNTRU}$ vector, then Section 27.4 shows that under some explicit conditions on the error terms $e_0, \ldots, e_\ell$, the shortest lattice vector is certainly smaller than the given bound. In Section 27.6, these observations lead to an elementary distinction criterion that solves the $\mathsf{(M)iNTRU}$ decision problem using ordinary LLL reduction. It is noteworthy that in Section 27.6.2 an exact success condition based on the underlying error distribution is filtered out and in case this condition is satisfied, a success probability of at least $1 - \frac{1}{\sqrt{\pi\ell}} 2^{-(\ell+1)}$ is obtained. Chapter 28 presents a similar attack that is slightly more general but relies on heuristic assumptions. In Chapter 29, the first attack is generalized to the MiNTRU decision problem, which asks to distinguish a matrix $\mathbf{A} := \left[\mathbf{S}^{-1} \times (\mathbf{G} - \mathbf{E}) \mod q\right] \in \mathbb{Z}_q^{n \times m}$

where $\mathbf{S} \in (\mathbb{Z}_q)^{n\times n}_{inv}$ is unknown, $\mathbf{E} \in \mathbb{Z}_q^{n\times m}$ is unknown, and $\mathbf{G}$ is known, from a matrix that is sampled uniformly at random in $\mathbb{Z}_q^{n\times m}$. Despite the consideration of stronger lattice reductions in Equation (29.16), Section 29.2 yields that the described attack is not strong enough to put cryptographic applications in danger. In Chapter 30, the hardness of the MiNTRU assumption is questioned by comparing it with the NTRU problem and the learning-with-errors assumption. Finally, Chapter 31 gives an outlook on potential research objectives for the MiNTRU assumption.

Part V: A Conditional Attack Against Functional Encryption Schemes

Part V outlines a conditional attack against the indistinguishability security notion of public-key functional encryption, that endangers a family of DDH-based functional encryption schemes for the bounded-norm inner-product functionality. Chapter 32 reviews the motivation and application of functional encryption – an ambitious cryptographic paradigm established to allow evaluations over encrypted data that reveal the evaluation in plain without leaking further information on the underlying plaintexts. Chapter 33 formally defines the corresponding notion in the public-key setting. First, Definition 33.1 defines a functionality $\mathcal{F} : \mathcal{K} \times \mathcal{M} \to \Sigma \cup \{\perp\}$ as a function over a key space $\mathcal{K}$ and a message space $\mathcal{M}$ that either outputs a valid element from Σ or a special error symbol $\perp$ not included in Σ. Second, Definition 33.2 defines a functional encryption scheme for a functionality $\mathcal{F}$ as a quadruple of algorithms: $\mathsf{FE.Setup}$ – generating the master public and master secret key, $\mathsf{FE.KDer}$ – encrypting evaluation keys, $\mathsf{FE.Enc}$ – encrypting messages, and $\mathsf{FE.Dec}$ – either outputting $\mathcal{F}(k, m)$ if a ciphertext and an encrypted key match or $\perp$ otherwise. In Chapter 34, the most important security notions of functional encryption schemes are listed, and, in Chapter 35, data-privacy is deepened with a focus on indistinguishability. In Chapter 36, a conditional attack against the indistinguishability of functional encryption schemes is devised. To do so, Section 36.1 highlights the particular role of the error symbol $\perp$ and analyzes how it can be represented in a functional encryption scheme. It is put forth that a functionality may be split into a function and an error trigger. Section 36.2 investigates how this error trigger can be implemented in practice and concludes that only the decryption procedure may be suitable for such an implementation. Having in mind the special role of the error trigger, Section 36.3 revises the functional encryption indistinguishability notion. It claims that ciphertexts of two challenge messages m_0, m_1 are indistinguishable, even if the decryption procedure can be used. However, as in this way ciphertexts can be evaluated, the notion restricts an adversary to use keys k such that $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1)$ only. Section 36.4 outlines that if $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1) = \perp$, then the decryption procedure can be hijacked to devise a conditional distinguisher. Essentially, it is noted that despite the same evaluation output, the decryption procedure may behave differently for such messages. Working out the underlying conditions culminates in Theorem 36.2. Chapter 37 motivates the previous study by showing that the attack partially invalidates a family of existing DDH-based functional encryption schemes for the inner-product functionality when restricting them to bounded-norm evaluations. In Chapter 38, a hypothetical indistinguishability definition, that would weaken security by forbidding our attack, is studied. It is concluded that such a definition would declare the previously mentioned functional encryption schemes secure for the bounded-norm inner-product functionality, but would be incompatible with other primitives. At last, in Chapter 39, some promising research questions on functional encryption are highlighted.

Special Thanks

Usually, scientific results do not emerge from a single mind, but they grow from the cumulative thoughts of the scientific community. Each expert grants a different perspective on a topic, which directly leads to scientific progress. Before we start to explore the contents of this dissertation, I would like to address some special thanks to those experts whose thoughts allowed me to advance my research. Under the slogan "*Cui honorem, honorem!*", I wish to point out the passages of the upcoming development that have been inspired, corrected, or improved by my peers.

Dissertation

First of all, I would like to thank my supervisor Dr Volker Müller for repeatedly proofreading my whole dissertation, improving its structure, and verifying my results. I also thank my non-supervising jury members Dr Wouter Castryck, Dr Nico Döttling, Dr Franck Leprévost, and Dr Gabor Wiese for correcting my thesis.

Part I: Toolbox

I thank Dr Christophe Ley for proofreading Chapter 2. I am indepted to Dr Gabor Wiese for proofreading Chapter 5 and Chapter 7 and simplifying the proof of Theorem 5.1. I thank Dr Luca Notarnicola and Dr Agnese Gini for their continuous help on Chapter 6. Furthermore, I thank Dr Agnese Gini for simplifying the proof of Lemma 7.2.

Part II: Simultaneous Chinese Remaindering

I thank Dr Volker Müller for suggesting this research topic. I also thank him for his comments on the algorithms in Section 8.4, for formalizing Definition 9.1, and for improving the algorithm in Section 9.3. I am indebted to

Reynaldo Gil Pons for pointing out Section 10.2.2 hinting that the Bounded Simultaneous Chinese Remainder Problem is $\mathsf{NP}-\mathsf{complete}$. This inspired Chapter 10. I thank Dr Vitor Pereira and Reynaldo Gil Pons for their improvement suggestions on Chapter 10. I thank Dr Franck Leprévost for correcting the algorithm in Section 13.2.1 and Dr Volker Müller for pointing out Remark 13.1. I am grateful for Dr Luca Notarnicola's suggestions on Chapter 15. I am indebted to Dr Gabor Wiese for providing me with the recursive counting method described in Section 18.2.2. I thank Dr Volker Müller for his suggestion on profiling the algorithm in Section 18.5 to make it more time-efficient.

Part III: A Conjecture On Primes In Arithmetic Progressions And Geometric Intervals

I thank Dr Franck Leprévost for his detailed correction of this part leading to an improved presentation. I thank Dr Gabor Wiese for pointing out the link of arithmetic progressions to Chebotarav's density theorem described in Section 20.4. I thank Dr Volker Müller for his help on the interpretation of Conjecture 21.2. I am grateful for Dr Franck Leprévost's suggestion to include Section 22.5.

Part IV: On The (M)iNTRU Assumption Over Finite Rings

I thank Dr Răzvan Roşie for reaching out to me to question a new computational hardness assumption. I am grateful that Dr Agnese Gini proofread Chapter 26 multiple times, until correct and consistent definitions were found. I thank Dr Volker Müller and Dr Gabor Wiese for their suggestions on the formalization of Chapter 26. I thank Dr Răzvan Roşie for his help on the applications descibed in Chapter 26. I thank Dr Volker Müller for providing me with the idea of the attack in Chapter 27. I am indepted to Dr Răzvan Roşie for the comparison in Chapter 30.

Part V: A Conditional Attack Against Functional Encryption Schemes

I thank Dr Răzvan Roşie and Dr Rajeev Anand Sahu for proofreading this part multiple times and their help on the background work in Chapter 32. I am indebted to both of them for confirming the attack by pointing out the schemes listed in Section 37.2. I also thank them for their insights on predicate encryption used in Section 38.2.

List of publications

The contents of the fourth part have been published as:

Jim Barthel, Volker Müller, and Răzvan Roşie (2021). On the MiNTRU assumption in the integer case. *ProvSec 2021*

The contents of the third part are going to be published as:

Jim Barthel and Volker Müller. (2022). A Conjecture on Primes in Arithmetic Progressions and Geometric Intervals, *Amer. Math. Monthly*

Other contributions are:

Jim Barthel, Marc Beunardeau, Răzvan Roşie, and Rajeev Anand Sahu (2021). Partitioned Searchable Encryption. *ProvSec 2021*

Jim Barthel and Răzvan Roşie (2021). NIKE from Affine Determinant Programs. *ProvSec 2021*

Contents

Affidavit **i**

Acknowledgements **iii**

Preface **v**

Special Thanks **xi**

List of publications **xiii**

Contents **xv**

I Toolbox **1**

Act I **3**

Abstract I **4**

Contents I **5**

1 Notations and conventions **7**

1.1 Set notations 7

1.2 Integers and real numbers 7

1.3 Other mathematical notations 8

1.4 Modulo operations 8

1.5 General conventions 9

1.6 Algorithmic notations 9

2 Statistics **11**

2.1 Probabilistic model 11

2.2 Probability . . . 12
2.3 Random variables . . . 13
2.4 Univariate distribution . . . 13
2.5 Multivariate distributions . . . 14
2.6 Particular distributions and sampling processes . . . 15

3 Complexity notions 17
3.1 Decision problems . . . 17
3.2 Solving methods . . . 18
3.3 Machine models . . . 19
3.3.1 Turing machines . . . 19
3.3.2 Deterministic Turing machines . . . 20
3.3.3 Non-deterministic Turing machines . . . 20
3.4 Reductions . . . 21
3.5 Complexity classes . . . 21
3.5.1 P . . . 21
3.5.2 NP . . . 21
3.5.3 NP-complete . . . 21
3.5.4 NP-hard . . . 22
3.5.5 P versus NP . . . 22
3.6 Remark on Boolean values . . . 22

4 3-SAT 23
4.1 Satisfiability problem . . . 23
4.2 Conjunctive normal form . . . 24
4.3 k-SAT . . . 25

5 Miscellaneous 27
5.1 Solving linear multivariate congruences . . . 27
5.2 Volume of a sphere of radius R . . . 28

6 Lattice preliminaries 33
6.1 Lattices . . . 33
6.2 Determinant . . . 33
6.3 Successive minima . . . 34
6.4 Minkowski's theorems . . . 34
6.5 Gaussian heuristic . . . 35
6.6 Sublattices . . . 35
6.7 Dual lattices . . . 36
6.8 Lattice reduction . . . 36

6.8.1 Gauss-Lagrange reduction 36
6.8.2 LLL 37
6.8.3 BKZ 37

7 Q-ary lattices 39
7.1 Definition and properties 39
7.2 Shortest vector approximation - Infinity norm 40
7.3 Shortest vector approximation - Euclidean norm 43

Bibliography I 47

II Simultaneous Chinese Remaindering 51

Act II 53

Abstract II 54

Contents II 55

8 Chinese Remaindering 59
8.1 The Chinese Remainder Problem 59
8.2 Complexity of Chinese Remaindering 60
8.2.1 Corresponding decision problems 60
8.2.2 The case of two congruences 61
8.2.3 The multi-congruence case 62
8.3 Chinese Remainder Theorem 63
8.4 Particular solving methods 63
8.4.1 Systematic search 64
8.4.2 Textbook CRT 64
8.4.3 Garner's algorithm 64
8.5 The power of Chinese Remaindering 65
8.6 Chinese Remaindering and Polynomial Interpolation 66
8.6.1 Lagrange 66
8.6.2 Newton 67

9 Simultaneous Chinese Remaindering 69
9.1 The Simultaneous Chinese Remainder Problem 69
9.2 Complexity of Simultaneous Chinese Remaindering 70
9.3 Recursive Garner 71

10 Two Simultaneous Chinese Remainder Decision Problems 73
10.1 Existential Simultaneous Chinese Remainder Problem 73
10.1.1 Proof intuition 74
10.1.2 Proof structure 74
10.1.3 A. Membership in NP 74
10.1.4 B. Problem construction 75
10.1.5 C. Solution matching 77
10.2 Bounded Simultaneous Chinese Remainder Problem 78
10.2.1 Proof intuition 78
10.2.2 Comment 79
10.2.3 Proof structure 79
10.2.4 A. Membership in NP 79
10.2.5 B. Problem construction 80
10.2.6 Bound analysis 82
10.2.7 Some observations 83
10.2.8 C. Solution matching 86
10.2.9 The hardness of different instances 87
10.2.10 A generalization 87

11 Simultaneous Chinese Remainder Problem variants 91
11.1 The Minimal Simultaneous Chinese Remainder Problem . . . 91
11.2 Other related problems 92
11.3 Problems in the literature 94
11.4 Applications 95
11.4.1 Factoring with extra information 95
11.4.2 Point counting on elliptic curves 96

12 An upper bound for the minimal solution 97
12.1 Intuition 97
12.2 Maximal number of solutions inside a given interval 98
12.3 Elementary upper bound 99
12.4 Optimal ordering 101
12.5 Limitations 102

13 Deterministic solving method 103
13.1 The trivial case: k=1 103
13.2 The first non-trivial case: k=2 103
13.2.1 A solving algorithm 104
13.2.2 Analysis 104
13.3 The general case 105

14 Family of minimizing functions 107

14.1 Declaring a new modulus and auxiliary coefficients 107

14.2 Constructing new functions minimizing in an extremum . . . 108

14.3 Intermediate sizes 112

14.4 A family of functions minimized in an extremum 113

15 Lattice solving methods 115

15.1 Coppersmith's method 115

15.1.1 The approach 116

15.1.2 Comments 117

15.2 Bleichenbacher-Nguyen method 118

15.2.1 First approach 118

15.2.2 Remarks 119

15.2.3 Second approach 120

16 Reformulations 125

16.1 A sieving problem 125

16.1.1 Intuition 125

16.1.2 Auxiliary notations 126

16.1.3 Elementary sieve results 127

16.1.4 Brun's sieve 129

16.2 A subset sum problem 130

16.2.1 The multiple-choice knapsack problem and its subset sum equivalent 131

16.2.2 Problem construction 131

16.2.3 Remarkable results 132

17 An illustrative example 133

18 Experiments and heuristics 135

18.1 Cumulative distribution 135

18.2 Fixed remainder set size 138

18.2.1 Closed form formula 138

18.2.2 Recursive counting function 139

18.2.3 A constructive approach 141

18.2.4 Improved counting method 144

18.2.5 Particular observations 146

18.3 Simulating the distributions 148

18.4 The maximal minimal solution 150

18.5 Remainder sets of size two 152

18.6 Moduli with common divisors 154

19 Open Questions 155

Bibliography II 157

III A Conjecture On Primes In Arithmetic Progressions And Geometric Intervals 161

Act III 163

Abstract III 164

Contents III 165

20 The distribution of primes 167
20.1 Primes in general 167
20.2 Primes in arithmetic progressions 168
20.3 Another prime counting function 169
20.4 A generalization of primes in arithmetic progressions 170

21 A new conjecture 171
21.1 Linnik's constant 171
21.2 A generalization of Linnik's constant 172

22 A partial proof 173
22.1 An elementary proof for progression size q=2 173
22.2 An existential proof for every sufficiently large exponent t . . 173
22.3 Consideration under ERH 175
22.4 Computational verification 177
22.5 Computational cost 178
22.6 Conclusion 179

23 Relation to other conjectures 181
23.1 Carmichael's conjecture 181
23.2 Computational verification 181
23.3 Pomerance's conjecture 182
23.4 The link with our conjecture 183

24 Open Questions 185

Bibliography III 187

IV On The (M)iNTRU Assumption Over Finite Rings 191

Act IV 193

Abstract IV 194

Contents IV 195

25 Introduction 197

26 Inhomogeneous NTRU assumptions 199
- 26.1 The inhomogeneous NTRU assumption 199
  - 26.1.1 The iNTRU assumption 199
  - 26.1.2 Applications 200
- 26.2 The matrix inhomogeneous NTRU assumption 202
  - 26.2.1 The MiNTRU assumption 202
  - 26.2.2 Applications 203
- 26.3 The (M)iNTRU problem 204

27 Attacking the (M)iNTRU assumption - First approach 205
- 27.1 Lattice construction 205
- 27.2 A lattice basis 206
- 27.3 Case of a random challenge vector 207
- 27.4 Case of a (M)iNTRU challenge vector 208
  - 27.4.1 First observation 208
  - 27.4.2 Second observation 209
  - 27.4.3 An unusually short lattice vector 209
- 27.5 Lattice reduction 210
  - 27.5.1 Case of a random challenge vector 210
  - 27.5.2 Case of a (M)iNTRU challenge vector 210
- 27.6 Conclusion 211
  - 27.6.1 Comment on the error distribution 211
  - 27.6.2 Comment on the success probability 211
- 27.7 Empirical tests 212

28 Attacking the (M)iNTRU assumption - Second approach 213
- 28.1 Lattice construction 213
- 28.2 Case of a random challenge vector 214

28.3 Case of a (M)iNTRU challenge vector 218
28.3.1 An observation 218
28.3.2 An unusually short lattice vector 218
28.4 Lattice reduction 219
28.4.1 Case of a random challenge vector 220
28.4.2 Case of a (M)iNTRU challenge vector 220
28.5 Conclusion 221
28.6 Empirical tests 221

29 Generalizations and limitations of our attacks 223
29.1 MiNTRU 223
29.1.1 Lattice construction 223
29.1.2 Case of a random challenge matrix 224
29.1.3 Case of a MiNTRU challenge matrix 225
29.1.4 Lattice reduction 226
29.1.5 Conclusion 228
29.1.6 Improvements and observations 228
29.2 Impact on cryptographic schemes 229

30 Comparisons with other hardness assumptions 231
30.1 NTRU 231
30.2 Learning-with-Errors 232
30.2.1 Multi-secret LWE 232
30.2.2 Hardness of multi-secret LWE 232
30.2.3 Connection to MiNTRU 233

31 Open Questions 235

Bibliography IV 237

V A Conditional Attack Against Functional Encryption Schemes 241

Act V 243

Abstract V 244

Contents V 245

32 A short review of functional encryption **247**
32.1 Historical motivation 247
32.2 Applications 248
32.3 Instantiations 248

33 Functional encryption **249**

34 Security notions **251**
34.1 Data-privacy 251
34.2 Function-hiding 252
34.3 Consistency and robustness 252

35 Focus on data-privacy **253**
35.1 Indistinguishability 253
35.2 Semantic security 255
35.3 Informal comparison 255

36 Our attack **257**
36.1 Functionalities and their error symbols 258
36.2 Functional encryption and error triggers 258
36.3 Revision of indistinguishability 259
36.4 Attacking indistinguishability 260
36.5 Summary 262

37 Impact on the bounded-norm inner-product functionality **263**
37.1 Bounded-norm inner-product functionality 263
37.2 Applicability of our attack 264
37.2.1 A concrete example 265
37.2.2 Our attack against other candidate constructions . . . 266

38 A patch for indistinguishability? **267**
38.1 A weaker indistinguishability notion 267
38.2 The shortcoming of this definition 268
38.2.1 Identity-based encryption 269
38.2.2 Attribute-based encryption 270
38.2.3 Incompatibility with predicate encryption 272

39 Open Questions **273**

Bibliography V **275**

Ending act **281**

Part I

Toolbox

Act I: The collection

“After many years of patience, the day has come”, thought Jay, “finally, you can help pa and pops to work on the family collection.” With great excitement, he got up and rushed to the kitchen. Standing next to the self-crafted oak wood table, he stared into an empty room and sighed: ”Where are they? Did they forget my birthday?” He heard a soft rustle in his back and turned around expecting a beloved one, but it was just Mrs Skizzles, granny’s cat, pleading for breakfast. Heartbroken, Jay decided to go back to sleep when his pa entered from the garage.

“You are already up?”, exclaimed his dad seemingly surprised, “your mom and Missy went out to buy pastries; they should be back every minute.” While handing Jay a wrapped box, he suggested: “In the meantime, you can open my present!”. Jay speculated that this may be his first personal collectable and remembered the very first time he looked at the most precious family item. As a kid, he discovered the shiny acquisition that sparkled in the sun putting a stream of light over the ceiling; a spectacular show for a four-year-old. A fine relief portrayed a man and sharp edges delimited an unintelligible inscription. Pops’ bedtime stories helped little Jay to understand the historical and cultural background of this strangely shaped golden platter. He got to know that this Byzantine coin portraying Justinian II is a masterpiece of coinage and he grew a lasting passion for numismatics.

Knowing that his pa promised to introduce him on the occasion of his fourteenth birthday to the family tradition, he expected a small set of coins. That is why he was stunned to find only a set of bizarre gadgets in his gift. “Won’t I contribute to the collection”, Jay queried disappointed. “You will”, assured his pa, “but first, you need to familiarize yourself with the tools!”.

Abstract I

This part is devoted to the preparation of a collection of results that will be needed in the remainder of the thesis. First, we establish some mathematical formalism and define our computer scientific notations. If not stated otherwise, the described conventions hold for the remainder of the document. Next, we introduce some statistical notions. Starting from random experiments, we define random variables, probabilities, and random vectors. Then, we present our complexity-theoretic framework. After characterizing decision problems and their solving methods, we discuss some machine models and revise the most important complexity classes. Subsequently, we zoom in on the first $\mathsf{NP}-\mathsf{complete}$ decision problem known as 3-SAT. Following this information theoretic deviation, we concentrate on some mathematical results. We compute the number of solutions of a linear multivariate congruence and then we approximate the volume of a high-dimensional sphere. Next, we focus on the geometry of numbers by introducing integer row lattices. We start by defining some of their invariants such as the determinant and their successive minima. Then, we announce Minkowski's theorems and discuss some lattice reduction techniques. After revising the most important properties of lattices, we have a look at so-called q-ary lattices. Those lattices are of particular interest in cryptography as they have some peculiar properties. Through a detailed analysis, we manage to develop two probabilistic lower bounds for the shortest vector in a q-ary lattice.

Contents I

Act I **3**

Abstract I **4**

Contents I **5**

1 Notations and conventions **7**

1.1 Set notations 7
1.2 Integers and real numbers 7
1.3 Other mathematical notations 8
1.4 Modulo operations 8
1.5 General conventions 9
1.6 Algorithmic notations 9

2 Statistics **11**

2.1 Probabilistic model 11
2.2 Probability 12
2.3 Random variables 13
2.4 Univariate distribution 13
2.5 Multivariate distributions 14
2.6 Particular distributions and sampling processes 15

3 Complexity notions **17**

3.1 Decision problems 17
3.2 Solving methods 18
3.3 Machine models 19
3.3.1 Turing machines 19
3.3.2 Deterministic Turing machines 20
3.3.3 Non-deterministic Turing machines 20
3.4 Reductions 21

3.5 Complexity classes . . . 21
3.5.1 P . . . 21
3.5.2 NP . . . 21
3.5.3 NP-complete . . . 21
3.5.4 NP-hard . . . 22
3.5.5 P versus NP . . . 22
3.6 Remark on Boolean values . . . 22

4 3-SAT 23
4.1 Satisfiability problem . . . 23
4.2 Conjunctive normal form . . . 24
4.3 k-SAT . . . 25

5 Miscellaneous 27
5.1 Solving linear multivariate congruences . . . 27
5.2 Volume of a sphere of radius R . . . 28

6 Lattice preliminaries 33
6.1 Lattices . . . 33
6.2 Determinant . . . 33
6.3 Successive minima . . . 34
6.4 Minkowski's theorems . . . 34
6.5 Gaussian heuristic . . . 35
6.6 Sublattices . . . 35
6.7 Dual lattices . . . 36
6.8 Lattice reduction . . . 36
6.8.1 Gauss-Lagrange reduction . . . 36
6.8.2 LLL . . . 37
6.8.3 BKZ . . . 37

7 Q-ary lattices 39
7.1 Definition and properties . . . 39
7.2 Shortest vector approximation - Infinity norm . . . 40
7.3 Shortest vector approximation - Euclidean norm . . . 43

Bibliography I 47

Chapter 1

Notations and conventions

1.1 Set notations

We use standard notations on sets where $\cap$ denotes intersection, $\cup$ denotes union, and $\setminus$ denotes exclusion. A set $\mathcal{S}$ can be defined by enumeration, delimiting its elements by curly brackets $\{\ \}$, or, if another set $\mathcal{S}'$ is given, by the set-builder notation $\mathcal{S} = \{n \in \mathcal{S}' \mid P(n)\}$ meaning that $\mathcal{S}$ contains all elements of $\mathcal{S}'$ that satisfy the property P. Set inclusion is denoted by $\subseteq$, where the inclusion may not be strict. The set product of two sets $\mathcal{S}_1$ and $\mathcal{S}_2$ is denoted by $\mathcal{S}_1 \times \mathcal{S}_2$. $\mathcal{S}^m$ denotes the set of row vectors with entries in $\mathcal{S}$ and $\mathcal{S}^{k\times m}$ denotes the set of matrices with k rows and m columns. We confound $\mathcal{S}^{1\times m}$ and $\mathcal{S}^m$. The infimum (inf), supremum (sup), minimum (min), and maximum (max) of a set $\mathcal{S}$ is either denoted by the subscript-element notation, e.g., $\min_{s\in\mathcal{S}} s$ or the set-input notation, e.g., $\min \mathcal{S}$. We abuse the notation by using it with lists, vectors, and matrices. The cardinality of $\mathcal{S}$ is given by $|\mathcal{S}|$.

1.2 Integers and real numbers

$\mathbb{N}$ stands for the natural numbers including 0, $\mathbb{Z}$ stands for the ring of integers and $\mathbb{R}$ denotes the ring of real numbers. Particular subsets of those rings can be obtained using subscript and superscript notation. For example $\mathbb{Z}_{>2}$ denotes the set of integers strictly larger than two and $\mathbb{R}_+$ denotes the set of non-negative real numbers. Intervals are delimited by brackets [ ] and parentheses ( ), where a bracket indicates that the corresponding interval bound is included and a parenthesis denotes the contrary. Given two integers a, b that are not both zero, $a|b$ means that a divides b, $\gcd(a, b)$ denotes the greatest common divisor and $\mathrm{lcm}(a,b)$ the lowest common multiple of a and b. Assuming $0 \leq a \leq b$, we denote by $b!$ the factorial of b and by $\binom{b}{a} = \frac{b!}{a!(b-a)!}$ the binomial coefficient of b by a. For a real value α, we indicate by $\lfloor \alpha \rfloor$ the maximal integer smaller than or equal to α, and by $\lceil \alpha \rceil$, the minimal integer larger than or equal to α. The exponential function is given by $\exp(\alpha)$ or e^{α}. The natural logarithm of $\alpha > 0$ is given by $\log(\alpha)$, and the base $b \in \mathbb{Z}_{\geq 2}$ logarithm is denoted by $\log_b(\alpha)$.

1.3 Other mathematical notations

Given a ring $\mathcal{R}$, we denote by $\mathcal{R}^{\times}$ the invertible elements in $\mathcal{R}$. To avoid any confusion, we set $(\mathcal{R})_{inv}^{m\times m}$ to be the set of invertible matrices with entries in $\mathcal{R}$. A norm on $\mathcal{R}$ is indicated by $\|\cdot\|$. On $\mathbb{R}$, $|\cdot|$ stands for the absolute value. For a matrix $\mathbf{A}$, we denote its transpose by $\mathbf{A}^{\top}$. a^{-1} denotes the inverse of a with respect to its ambient space. $\langle\cdot,\cdot\rangle$ stands for an inner-product function. Given a function f, we mean by $\mathrm{im}(f)$ its image and by $\ker(f)$ its kernel. Real number approximations are given by $\approx$ and group isomorphisms are indicated by $\simeq$. Asymptotic approximations are given by $\sim$ or the Landau symbols (see Section 3.2). Equality is given by $=$, but the first time a symbol is defined, we use $:=$.

1.4 Modulo operations

Let $q \in \mathbb{Z}_{\geq 2}$. Then, $\mathbb{Z}/q\mathbb{Z}$ denotes the ring of integers modulo q. We define $\mathbb{Z}_q := \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ and note that $\mathbb{Z}/q\mathbb{Z}$ is in bijection with $\mathbb{Z}_q$. For $a, b \in \mathbb{Z}$, we distinguish three distinct modular operations:

1. $a \equiv b \mod q$ denotes that $[a] = [b]$ in $\mathbb{Z}/q\mathbb{Z}$.
2. $[a \mod q]$ denotes the unique integer $a_0 \in \mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ such that $a \equiv a_0 \mod q$.
3. $[\![a \mod q]\!]$ denotes the unique integer $a_0 \in \mathbb{Z} \cap [0, q)$ such that $a \equiv a_0 \mod q$.

We abuse the notation and use the same symbolism for vectors and matrices where the operations are carried out componentwise. For example, if $\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{Z}^n$, then $[\mathbf{v} \mod q] = ([v_1 \mod q], \ldots, [v_n \mod q])$. Operations inside brackets, such as $+, -, \cdot$, or inverses always take place modulo q. For example $[a + b \cdot s^{-1} \mod q]$ first computes $a + b \cdot s^{-1}$ modulo q

where s^{-1} denotes the inverse of s modulo q, and subsequently outputs the unique representative in $\mathbb{Z}_q$. We let $\mathbb{Z}_q^{\times}$ denote the set of invertible elements of $\mathbb{Z}_q$ modulo q. We note that $x \in \mathbb{Z}$ is invertible modulo q if and only if $\gcd(x, q) = 1$. We let $(\mathbb{Z}_q)_{inv}^{n \times n}$ denote the set of matrices in $\mathbb{Z}_q^{n \times n}$ that are invertible modulo q.

1.5 General conventions

Usually, lowercase letters such as a stand for elements. Bold lowercase letters like $\mathbf{a}$ denote vectors and $\mathbf{a}_i$ denotes the i-th entry of the vector $\mathbf{a}$. Bold uppercase letter, such as $\mathbf{A}$, denote matrices and $\mathbf{A}_{i,j}$ denotes the component in the i-th row and j-th column of the matrix. Normal uppercase and calligraphic uppercase letters stand for sets.

1.6 Algorithmic notations

Algorithms are seen as being randomized and modeled by a Turing machine. The action of running an algorithm $\mathcal{A}$ on an input input with access to a subroutine (or oracle) $\mathcal{O}$ is denoted by $\mathcal{A}^{\mathcal{O}(\cdot)}(\mathsf{input})$. The assignment of an algorithm output is denoted by $\leftarrow$. If the algorithm is deterministic, then the plain arrow is used. If the algorithm is randomized, then the arrow is indexed by \$. For example, the action of running a randomized algorithm $\mathcal{A}$ with a subroutine $\mathcal{O}$ on input input with a uniformly at random sampled random coins r and assigning the output to output is denoted by $\mathsf{output} \leftarrow_{\$} \mathcal{A}^{\mathcal{O}(\cdot)}(\mathsf{input}; r)$. If the random coin is implicit or unknown, it may be removed from the notation. We abuse the arrow notation for value assessments. Deterministic value assessments are denoted by $\leftarrow$. For illustration, $a \leftarrow 2$ means that a receives the value 2. Sampling a value uniformly at random from a finite set $\mathcal{S}$ and assessing it to a variable a is denoted by $a \leftarrow_{\$} \mathcal{S}$. If another distribution is used for the sampling process, say χ, then we write $a \leftarrow_{\chi} \mathcal{S}$ (see also Chapter 2). We denote the set of all functions that are negligible with respect to a parameter λ by $\textsc{Negl}(\lambda)$. The bit size of an integer n is the minimal positive integer β such that $|n| < 2^{\beta}$.

Chapter 2

Statistics

The upcoming presentation is based on [Shy13, HKO01], but considers a finite discrete setting.

2.1 Probabilistic model

A probabilistic model is a mathematical description of an uncertain situation. A probabilistic model involves an experiment that produces exactly one out of several possible outcomes which cannot be predicted before carrying out the experiment. The set of all possible outcomes of an experiment is completely determined before it is carried out and the experiment can be repeated under the same conditions as often as desired. An experiment can be anything that may result in a different outcome in each iteration, for example, a coin toss or radiation counts. The set of all possible outcomes Ω is called the *sample space* and a subset $A \subseteq \Omega$ is called an *event*.

A *σ-field* $\mathcal{F}$ over Ω is a non-empty set of events that satisfies the following conditions:

- $\Omega \in \mathcal{F}$.
- If $A \in \mathcal{F}$, then $\Omega \setminus A \in \mathcal{F}$.
- For every collection of events $\{A_i \in \mathcal{F}\}_{i \in \mathcal{N}}$ where $\mathcal{N} \subseteq \mathbb{N}$, we have $\bigcup_{i \in \mathcal{N}} A_i \in \mathcal{F}$.

These conditions imply that $\emptyset \in \mathcal{F}$ and that $\bigcap_{i \in \mathcal{N}} A_i \in \mathcal{F}$. For a finite or infinite countable sample set Ω, a σ-field is given by the *powerset* of Ω denoted by $\mathcal{P}(\Omega)$ and including all subsets of Ω. For $\Omega = \mathbb{R}$, a σ-field is given by the *Borel set* denoted by $\mathcal{B}(\mathbb{R})$ and generated by the open intervals of $\mathbb{R}$.

We note that $\mathcal{B}(\mathbb{R})$ includes every open, closed, and semi-closed interval in $\mathbb{R}$, as well as singleton sets and intervals tending to infinity. A *measure* on a σ-field $\mathcal{F}$ is any mapping $\tau : \mathcal{F} \longrightarrow \mathbb{R}$ such that

- $\tau(A) \geq 0$ for all $A \in \mathcal{F}$,
- $\tau(\emptyset) = 0$, and
- for every collection of disjoint events $\{\mathcal{A}_i\}_{i \in \mathcal{N}}$ where $\mathcal{N} \subseteq \mathbb{N}$, that is $A_i \cap A_j = \emptyset$ for all $i \neq j$, we have $\tau\left(\bigcup_{i \in \mathcal{N}} \mathcal{A}_i\right) = \sum_{i \in \mathcal{N}} \tau(\mathcal{A}_i)$.

If $\mathcal{F}$ is a σ-field over Ω and a measure on $\mathcal{F}$ exists, then we call $\mathcal{F}$ a *measurable event space* over Ω. A remarkable measure on $\mathcal{B}(\mathbb{R})$ is the *Lebesgue* measure defined by $L([a,b]) = b - a$ for $a < b$.

2.2 Probability

A *probability measure* $\mathbb{P}$ on a *σ-field* $\mathcal{F}$ over a sample space Ω is any mapping $\mathbb{P} : \mathcal{F} \longrightarrow [0,1]$ that satisfies:

- $\mathbb{P}(\mathcal{A}) \geq 0$ for every event $\mathcal{A} \in \mathcal{F}$,
- $\mathbb{P}(\Omega) = 1$, and
- for every collection of disjoint events $\{\mathcal{A}_i\}_{i \in \mathcal{N}}$ where $\mathcal{N} \subseteq \mathbb{N}$, we have $\mathbb{P}\left(\bigcup_{i \in \mathcal{N}} \mathcal{A}_i\right) = \sum_{i \in \mathcal{N}} \mathbb{P}(\mathcal{A}_i)$.

A probability measure $\mathbb{P}$ is indeed a measure as $\mathbb{P}(\Omega) = \mathbb{P}(\Omega) + \mathbb{P}(\emptyset)$, such that $\mathbb{P}(\emptyset) = 0$. Furthermore, it has many nice properties. The *complementary probability* rule states that $\mathbb{P}(\Omega \setminus A) = 1 - \mathbb{P}(A)$ for all $A \subseteq \Omega$. The *conditional probability rule* yields the probability of an event $A \in \mathcal{F}$ given an event $B \in \mathcal{F}$ as $\mathbb{P}(A|B)\mathbb{P}(B) = \mathbb{P}(A \cap B)$. Furthermore, two events $A, B \in \mathcal{F}$ are called *independent* if $\mathbb{P}(A|B) = \mathbb{P}(A)$. A *probability space* is any triplet $(\Omega, \mathcal{F}, \mathbb{P})$ where Ω is a sample space, $\mathcal{F}$ is a σ-field over Ω called the *event space*, and $\mathbb{P}$ is a probability measure over $\mathcal{F}$. One example of a probability space is $(\Omega, \mathcal{P}(\Omega), \mathbb{P}_\Omega)$ where Ω is finite and $\mathbb{P}_\Omega(A) = \frac{|A|}{|\Omega|}$ for all $A \subseteq \Omega$. Another example is $([0,1], \mathcal{B}([0,1]), \mathbb{P}_L)$ where $\mathbb{P}_L([a,b]) = b - a$ is the Lebesgue measure on $[0,1]$.

2.3 Random variables

For $i \in \{1, 2\}$, let $\mathcal{F}_i$ be a measurable event space over Ω_i. A function $g : \Omega_1 \to \Omega_2$ is a *measurable function* if the preimage of g satisfies $g^{-1}(B) = \{a \in \Omega_1 \mid g(a) \in B\} \in \mathcal{F}_1$ for all $B \in \mathcal{F}_2$. We note that g is defined for all the elements in the sample space Ω_1 and not for the events in $\mathcal{F}_1$. If Ω_1 is countable and $\mathcal{F}_1 = \mathcal{P}(\Omega)$, then any function from Ω_1 to Ω_2 is measurable. Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we define a *real-valued random variable* over Ω as a measurable function $\omega : \Omega \longrightarrow \mathbb{R}$ with respect to $\mathcal{B}(\mathbb{R})$. Since ω is measurable, any interval in $\mathcal{B}(\mathbb{R})$ must have an inverse image in $\mathcal{F}$. For any borelian $B \subseteq \mathbb{R}$, we define a probability measure on $\mathcal{B}(\mathbb{R})$ by setting $\mathbb{P}_\omega(B) := \mathbb{P}(\omega^{-1}(B)) = \mathbb{P}(\{a \in \Omega \mid \omega(a) \in B\})$. We abuse the notation and set $\mathbb{P}(\omega \in \mathbb{B}) := \mathbb{P}_\omega(B)$ which intuitively means that the outcome of ω is in B. Similarly, for $b \in \mathbb{R}$, we define $\mathbb{P}(\omega = b) := \mathbb{P}_\omega(\{b\})$, and we set $\mathbb{P}(\omega \leq b) := \mathbb{P}_\omega((-\infty; b])$. We proceed in the same way for other comparison symbols. This abuse of notation is motivated by the fact that $\mathbb{P}_\omega$ behaves like $\mathbb{P}$, but relies on $\mathbb{R}$ only. The *support* of a random variable ω is the set of real numbers $b \in \mathbb{R}$ such that $\mathbb{P}(\omega \leq b+\epsilon) > \mathbb{P}(\omega \leq b-\epsilon)$ for all $\epsilon > 0$. If a real-valued random variable has a finite support, then it is called *finite*. In the following, we consider finite real-valued random variables only.

2.4 Univariate distribution

Let $\omega : \Omega \longrightarrow \mathbb{R}$ be a finite real-valued random variable over the sample space Ω with support E. The *cumulative distribution function* $F_\omega : \Omega \longrightarrow [0, 1]$ of ω is defined by $F_\omega(b) := \mathbb{P}(\omega \leq b)$ for all $b \in \mathbb{R}$. The cumulative distribution function is completely defined by the *probability mass function* $p_\omega : \Omega \longrightarrow\ [0, 1]$ of ω defined by $p_\omega(b) := \mathcal{P}(\omega = b)$. Indeed, for $b \in \mathbb{R}$, let $E_{\leq b} = E \cap (-\infty, b]$, then $F_\omega(b) = \sum_{b' \in E_{<b}} p_\omega(b')$. Reciprocally, the cumulative distribution function of ω completely determines the probability mass function of ω by $p_\omega(b) = F_\omega(b)$ if $b \leq \min(E)$ and $p_\omega(b) = F_\omega(b) - F_\omega(b_0)$ where $b_0 = \max E_{<b}$. The *frequency distribution* of ω is a table that displays the frequency of each outcome of ω. It is clear that the probability mass function and the frequency distribution of ω are equivalent.

The *expectation*, or *mean*, of a finite real-valued random variable ω with support E is defined by $\mu_\omega := \mathbb{E}[\omega] := \sum_{b \in E}\ b\ p_\omega(b)$. Furthermore, $\mathbb{E}[a\omega + b] = a\mathbb{E}[\omega] + b$ for all $a, b \in \mathbb{R}$. We note that if $g : \mathbb{R} \to \mathbb{R}$, then $g \circ \omega$ is a finite real-valued random variable over Ω with probability mass function $p_{g\circ\omega}(b) = \mathbb{P}(g \circ \omega = b)$ for all $b \in \mathbb{R}$. Additionally, $\mathbb{E}[g \circ \omega] = \sum_{b \in E} g(b) p_\omega(b)$.

The *variance* of ω is defined by $\sigma_\omega^2 := \mathbb{E}[\omega^2] - \mathbb{E}[\omega]^2$. The *standard deviation* is defined by $\sigma_\omega := \sqrt{\mathbb{E}[\omega^2] - \mathbb{E}[\omega]^2}$. Chebychev's inequality states that for a random variable ω with finite expected value μ and finite non-zero variance σ^2, we have $\mathbb{P}(|\omega - \mu| \leq k\sigma) \leq \frac{1}{k^2}$ for every $k > 0$. This indicates in particular that the probability that ω outputs values outside the interval $[\mu - 10\sigma, \mu + 10\sigma]$ is only $\frac{1}{100}$. For $0 < \alpha < 1$, an α-quantile q_α of ω is any real number such that $\mathbb{P}(\omega \leq q_\alpha) \geq \alpha$ and $\mathbb{P}(\omega \geq q_\alpha) \geq 1 - \alpha$. If the set Q of all α-quantiles is given by $Q = [b_1, b_2]$ for some consecutive $b_1 < b_2 \in E$, then we define the α-quantile as $\frac{b_1+b_2}{2}$. The median is the $\frac{1}{2}$-quantile, the first quartile is the $\frac{1}{4}$-quantile, and the third quartile is the $\frac{3}{4}$-quantile.

2.5 Multivariate distributions

A real-valued *random (row) vector* $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ over Ω is a vector whose coordinates are real-valued random variables over Ω. We note that the structure of Ω is not relevant. Equipping $\mathbb{R}^n$ with the lexicographic order, we can directly generalize the univariate case to this multivariate setting. Concretely, for $\mathbf{b} \in \mathbb{R}^n$, the *cumulative distribution function* for $\boldsymbol{\omega}$ is defined by $F_{\boldsymbol{\omega}}(\mathbf{b}) = \mathcal{P}(\boldsymbol{\omega} \leq \mathbf{b})$, the *probability mass function* of $\boldsymbol{\omega}$ by $p_{\boldsymbol{\omega}}(\mathbf{b}) = \mathcal{P}(\boldsymbol{\omega} = \mathbf{b})$, and the *frequency distribution* by the frequency of each possible outcome of $\boldsymbol{\omega}$. Hereinafter, we mean by distribution any of those representations as the others can be obtained from it. Random matrices can be defined similarly and generalize random vectors. The coordinates of a random vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_n)$ over Ω are said to be *independent* if $F_{\boldsymbol{\omega}}(\mathbf{b}) = \prod_{i=1}^n F_{\omega_i}(b_i)$ for all $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{R}^n$. A random vector is *finite* if it consists of finite random variables. The *support* of a finite real-valued random vector $\boldsymbol{\omega}$ is the set $E = \{\mathbf{b} \in \mathbb{R}^n \mid p_{\boldsymbol{\omega}}(\mathbf{b}) > 0\}$.

Expectations can be generalized to finite real-valued random vectors by setting $\mathbb{E}[\boldsymbol{\omega}] = \sum_{\mathbf{b} \in E} \mathbf{b} p_{\boldsymbol{\omega}}(\mathbf{b})$ where E denotes the support of $\boldsymbol{\omega}$. The resulting vector is the *mean vector* which is generally denoted by $\mu_{\boldsymbol{\omega}}$.

- (Linearity) If $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_m$ denote random vectors over Ω and $a_1, \ldots, a_m$ denote scalars, then $\mathbb{E}\left[\sum_{i=1}^m a_i \boldsymbol{\omega}_i\right] = \sum_{i=1}^m a_i \mathbb{E}[\boldsymbol{\omega}_i]$.

- (Linear transformation) If $\boldsymbol{\omega}$ denotes a random vector over Ω and $\mathbf{A} \in \mathbb{R}^{n \times m}$, then $\mathbb{E}\left[\boldsymbol{\omega}\mathbf{A}\right] = \mathbb{E}[\boldsymbol{\omega}]\mathbf{A}$.

- (Composition) Let $g : \mathbb{R}^n \longrightarrow \mathbb{R}^m$. Then $g \circ \boldsymbol{\omega}$ is a finite real-valued random vector and $\mathbb{E}[g \circ \boldsymbol{\omega}] = \sum_{\mathbf{b} \in E} g(\mathbf{b}) p_{\boldsymbol{\omega}}(\mathbf{b})$.

For a random vector $\boldsymbol{\omega} = (\omega_1, \dots, \omega_n)$ over Ω, the $n \times n$ matrix $\mathbf{R}_{\boldsymbol{\omega}} = \mathbb{E}[\boldsymbol{\omega}^\top \boldsymbol{\omega}]$ is called the *auto-correlation* matrix. The auto-correlation matrix is always symmetric, positive semi-definite, and its eigenvalues are real and non-negative. The $n \times n$ matrix $\mathbf{C}_{\boldsymbol{\omega}} = \mathbb{E}[(\boldsymbol{\omega} - \mu_{\boldsymbol{\omega}})^\top (\boldsymbol{\omega} - \mu_{\boldsymbol{\omega}})]$ is called the *auto-covariance* matrix and represents on its (i, j)-th entry the covariance between ω_i and ω_j. The entries on the diagonal represent the *variance* of the random variables $\omega_1, \dots, \omega_n$. We note that if an entry of $\mathbf{C}_{\boldsymbol{\omega}}$ is 0, then the corresponding random variables are said to be *uncorrelated*. Independent random variables are always uncorrelated, but the reciprocal does not necessarily hold. Through linear transformations, we observe that $\mathbf{C}_{\boldsymbol{\omega}} = \mathbf{R}_{\boldsymbol{\omega}} - \mu_{\boldsymbol{\omega}}^\top \mu_{\boldsymbol{\omega}}$.

2.6 Particular distributions and sampling processes

We call a distribution of a real-valued random vector $\boldsymbol{\omega}$ *symmetric* if for all $\mathbf{b} \in \mathbb{R}^n$, we have $p_{\boldsymbol{\omega}}(\mathbf{b}) = p_{\boldsymbol{\omega}}(-\mathbf{b})$. In this case, the mean is $\mu_{\boldsymbol{\omega}} = (0, \dots, 0)$, and, by Chebychev's inequality, we can expect the entries of a vector returned by $\boldsymbol{\omega}$ to not be much larger in absolute value than the standard deviation of the corresponding random variable. For a finite real-valued random variable with support E, we say that $\boldsymbol{\omega}$ *follows the uniform distribution* over E if $\mathcal{P}(\boldsymbol{\omega} = \mathbf{b}) = \frac{1}{|E|}$ for all $\mathbf{b} \in E$ and $\mathcal{P}(\boldsymbol{\omega} = \mathbf{b}) = 0$ otherwise. We say that a second random variable $\boldsymbol{\omega}'$ follows the distribution $\boldsymbol{\omega}$ over $\mathbb{R}$ if $\mathcal{P}(\boldsymbol{\omega}' = \mathbf{b}) = \mathcal{P}(\boldsymbol{\omega} = \mathbf{b})$ for all $\mathbf{b} \in \mathbb{R}$. We refer to [Shy13, HKO01] for a broad overview of the most relevant distributions.

Let $\boldsymbol{\omega}$ be a finite real-valued random vector with support $E \subseteq \mathbb{R}$. We abuse our notation and call any distribution of $\boldsymbol{\omega}$ *the* distribution $\boldsymbol{\omega}$. Furthermore, we say that a vector $\mathbf{v} \in \mathbb{R}$ has been sampled following the distribution $\boldsymbol{\omega}$ if $\mathbf{v}$ is an output of $\boldsymbol{\omega}$. Intuitively, it is a vector sampled from the support of $\boldsymbol{\omega}$ such that the probability of obtaining $\mathbf{v}$ is $p_{\boldsymbol{\omega}}(\mathbf{v})$. We fix the notation $\mathbf{v} \leftarrow_{\boldsymbol{\omega}} \mathbb{R}$ to denote that $\mathbf{v} \in \mathbb{R}$ has been sampled following the distribution $\boldsymbol{\omega}$. If $\boldsymbol{\omega}$ follows the uniform distribution over E, we note $\mathbf{v} \leftarrow_{\$} E$.

If we have some information on the support of $\boldsymbol{\omega}$, we change our notation to indicate this. For example, if $E \subseteq E' \subseteq \mathbb{R}^n$, then we write $\boldsymbol{\omega} : \Omega \longrightarrow E'$ to indicate that values in $\mathbb{R}^n \setminus E'$ are impossible events. If a random variable with codomain $\mathbb{R}^n$ is desired, one can simply extend $\boldsymbol{\omega}$ by setting $p_{\boldsymbol{\omega}}(\mathbf{b}) = 0$ for all $\mathbf{b} \in \mathbb{R}^n \setminus E'$. We apply the same strategy for the other notations, such as $\mathbf{v} \leftarrow_{\boldsymbol{\omega}} E'$. Furthermore, we highlight that the above notions can be generalized to another codomain Ω' instead of $\mathbb{R}$. In this case, one generally speaks of a *random element*.

Chapter 3

Complexity notions

This chapter revises some elementary notions from complexity theory. The upcoming development is strongly based on [Lee90] and loosely based on [Coo00].

3.1 Decision problems

A *problem* is a set $X = X_I \times X_A$ of ordered pairs (I, A) of strings in $\{0,1\}^*$ representing all possible finite strings made of 0's and 1's, where $I \in X_I$ is called the problem *instance* and $A \in X_A$ is called an *answer* to that instance. A problem instance needs to be completely defined, meaning that for each instance $I \in X_I$ there needs to be at least one answer $A \in X_A$. This yields a total relation on X. A problem *function* $f = f_I \times f_A \subseteq X$ is a string relation in which each string $I \in f_I$ is the instance of precisely one problem. In other words, for each $I \in f_I$, there exists exactly one answer $A \in f_A$. A *decision* problem is a problem function

$$d = d_I \times d_A \subseteq \{0,1\}^* \times \{0,1\} \tag{3.1}$$

in which the only possible answers are 1 indicating "yes" and 0 indicating "no". As the answer values are fixed, the decision problems only differ in their instance values $d_I \subseteq \{0,1\}^*$. A particular subset L of $\{0,1\}^*$ is called a *language*. Its *complementary language* is $\mathsf{co}{-}L = \{0,1\}^* \setminus L$. A decision problem $d = d_I \times \{0,1\}$ is completely defined by its language $d_I = L$ and its affirmative solutions $A = 1$. To be precise, if L is a language, then the decision problem d_L corresponding to L is defined by

$$d_L = \{(x,1) : x \in L\} \cup \{(x,0) : x \in \mathsf{co}{-}L\}. \tag{3.2}$$

Reciprocally, if d is a decision problem, then the language $L(d)$ corresponding to $d = d_I \times \{0, 1\}$ is defined by

$$L(d) = \{x \in d_I : (x, 1) \in d\}. \tag{3.3}$$

We note that an instance set d_I can always be extended over all bit strings in $\{0, 1\}^*$ by including $(x, 0)$ for all $x \notin d_I$.

3.2 Solving methods

We regard a decision problem $d = d_I \times \{0, 1\}$ as *solved* if and only if there is a general method M called *solving method* that is able to solve any instance $I \in d_I$. In other words, M is an algorithm which on input $I \in d_I$ computes $A \in \{0, 1\}$ such that $(I, A) \in d$. The *size* of a problem instance I, denoted by $|I|$, is its length as a bit-string. Let M be a solving method of a decision problem d and let R be a set of resources used by that method. We define $R_M : \mathbb{N} \to \mathbb{N}$ by setting $R_M(n)$ to be the maximal amount of resources R, used when M is applied to any input x of size n. Common resources of interest are the *computational time* needed to solve a problem instance and the required *memory space*. As it is often complicated to assess the exact amount of resources, approximate bounds are used. We use the *Landau* symbols. Indeed, let f, g be two real-valued functions defined on some unbounded subset of positive real numbers such that $g(x) > 0$ for sufficiently large x:

1. We write $f(x) = o(g(x))$ if for all $C > 0$ there exists $x_0 \in \mathbb{R}$ such that $|f(x)| \leq Cg(x)$ for all $x \geq x_0$.

2. We write $f(x) = O(g(x))$ if there exist $C > 0$ and $x_0 \in \mathbb{R}$ such that $|f(x)| \leq Cg(x)$ for all $x \geq x_0$.

3. We write $f(x) = \Omega(g(x))$ if $\limsup\limits_{x \to +\infty} \left|\frac{f(x)}{g(x)}\right| > 0$.

The *requirements* of a problem X for resources from a resource set R under methods from a method class C is *upper bounded by* $T(n)$ if and only if there is a method $M \in C$ for solving X such that $R_M(n) = O(T(n))$ and it is *lower bounded by* $T(n)$ if and only if any method $M \in C$ to solve X satisfies $R_M(n) = \Omega(T(n))$. We have the following classification of requirements where $k > 1$ and $0 < c < 1$ denotes any positive constant:

Name	$R_M(n)$	Name	$R_M(n)$
Constant	$O(1)$	Cubic	$O\left(n^3\right)$
Log-logarithmic	$O(\log(\log(n)))$	Polynomial	$O\left(n^k\right)$
Logarithmic	$O(\log(n))$	Quasipolynomial	$O\left(n^{(\log(n))^k}\right)$
Polylogarithmic	$O\left((\log(n))^k\right)$	Subexponential	$O\left(2^{n^c}\right)$
Linear	$O(n)$	Exponential	$O\left(2^{n^k}\right)$
Linearithmic	$O(n\log(n))$	Factorial	$O(n!)$
Quasilinear	$O\left(n(\log(n))^k\right)$	Superexponential	$O\left(2^{2^{n^k}}\right)$
Quadratic	$O\left(n^2\right)$	Unbounded	$< \infty$

Figure 3.1: Classification of asymptotic upper bounds.

3.3 Machine models

To compare different solving methods, one needs to formalize their functioning. To do so, we define a list of allowed operations, which is called a *computation model*.

3.3.1 Turing machines

Intuitively, a *Turing machine* manipulates symbols, such as 0 and 1, on a strip of tape according to a predefined set of rules. In the abstract model, the machine operates on an infinite memory tape, which is divided into discrete *cells* that can be assigned a specific symbol. For each operation, the machine positions its *head* over one such cell and *reads* the symbol in the cell. Based on the scanned symbol and the current state of the machine that is registered in a user-specified finite table of instructions, the machine

1. either erases, writes, or updates a symbol in the cell,

2. moves either to the cell to the left of the current cell, moves to the cell to the right of the current cell, or stays at the current cell,

3. updates, if required, its state, and,

4. based on the symbol in the new cell and its new state, proceeds to another instruction or halts the computation.

Using this syntax, the *computation time* is defined to be the number of steps made before the machine halts and the *computation space* is defined to be the number of cells that are visited by the head during the whole computation. The computation space may be relatively small compared to the input size, but the computation time is at least as large as the input size. In practice, the strip of tape in a Turing machine might be split into multiple components, for example, we consider the following subdivision:

1. a semi-infinite read-only tape for input,

2. a semi-infinite write-only tape for output,

3. a read-write work tape.

3.3.2 Deterministic Turing machines

A *deterministic* Turing machine (DTM) is a Turing machine with the additional requirement that it needs to follow a fully deterministic instruction set. In other words, for each given state and cell input, the machine can only follow a single instruction. Such a machine is said to have solved a given problem d, if, whenever it starts with a problem instance I written in the leftmost cells of its input tape, it eventually halts with an answer A written in the leftmost cells of the output tape.

3.3.3 Non-deterministic Turing machines

A *non-deterministic* Turing machine (NDTM) is a Turing machine following a non-deterministic instruction set. Put differently, given a state and a cell input, the machine has multiple choices for its next move. As it is not clear which possibility will be chosen, its computation steps need to be seen as a tree, where the nodes and leaves correspond to the tape configurations that can be obtained. A non-deterministic Turing machine answers “yes” to a given problem instance I if the tree of reachable configurations contains any configuration in which the machine halts and the output tape contains the string representing “yes”. Otherwise, it answers “no”. In case the configuration tree is finite (we will only consider such cases hereinafter), the computation time corresponds to the depth of the tree and the computation space is the maximal number of cells used by any configuration.

3.4 Reductions

Let $X = X_I \times \{0,1\}$ and $Y = Y_I \times \{0,1\}$ be decision problems. A (Turing) *reduction* or *transformation* from X to Y is any function

$$f : \{0,1\}^* \longrightarrow \{0,1\}^* \tag{3.4}$$

that can be computed by a deterministic Turing machine, such that a problem instance $x \in X_I$ has the affirmative answer 1 if and only if the problem instance $y = f(x) \in y_I$ has the affirmative answer 1. We usually focus on *polynomial time reductions*, meaning that the function in the above definition needs to be computable in polynomial time.

3.5 Complexity classes

Complexity classes help us to classify problems based on their best known solving method. Hereinafter, we highlight some of the most relevant complexity classes and describe some of their inherent properties.

3.5.1 P

P is the complexity class that represents the set of all decision problems that can be solved in polynomial time by a deterministic Turing machine.

3.5.2 NP

NP is the complexity class of all decision problems whose instances with affirmative answer 1 allow a *witness* that can be verified in polynomial time. The computation time for this witness may be unbounded, as long as the verification can be carried out sufficiently fast. Equivalently, NP is the set of all decision problems that are solvable in polynomial time by a nondeterministic Turing machine.

3.5.3 NP-complete

$\mathsf{NP} - \mathsf{complete}$ is the complexity class containing all NP problems to which NP problems are reducible to in polynomial time. Put differently, a decision problem $X = X_I \times \{0,1\}$ is in $\mathsf{NP} - \mathsf{complete}$ if X is in NP and if any other decision problem $Y = Y_I \times \{0,1\}$ in NP can be reduced to X in polynomial time. Thus, if a deterministic polynomial-time algorithm can be found to solve any $\mathsf{NP} - \mathsf{complete}$ problem, then, every problem in NP can be solved in polynomial time.

3.5.4 NP-hard

$\mathsf{NP-hard}$ is the complexity class containing all problems to which every $\mathsf{NP-complete}$ problem can be reduced to in polynomial time. Put differently, a problem $X = X_I \times X_A$ is in $\mathsf{NP-hard}$ if there is a problem $Y = Y_I \times \{0, 1\}$ in $\mathsf{NP-complete}$ that can be reduced to X in polynomial time. Intuitively, these problems are at least as hard as the $\mathsf{NP-complete}$ problems. Contrary to $\mathsf{NP-complete}$ problems, $\mathsf{NP-hard}$ problems do not need to be in NP, nor to be decision problems. As all NP problems are reducible in polynomial time to all $\mathsf{NP-complete}$ problems, and all $\mathsf{NP-complete}$ problems are reducible in polynomial time to $\mathsf{NP-hard}$ problems, solving a single $\mathsf{NP-hard}$ problem in polynomial time yields a polynomial time solving method for all NP-problems. We note that any $\mathsf{NP-complete}$ problem is also $\mathsf{NP-hard}$.

3.5.5 P versus NP

The most famous open problem in computer science and a millennium problem of the Clay Mathematics Institute [CMI00] asks whether any decision problem whose proof can be verified in polynomial time can also be solved in polynomial time. It is straightforward that P is a subset of NP. The open question is thus whether NP problems have deterministic polynomial-time solutions.

3.6 Remark on Boolean values

To simplify some passages, we freely switch between the integer values $0, 1$ and the Boolean values $0, 1$.

Chapter 4

3-SAT

One of the most important complexity classes is the $\mathsf{NP-complete}$ class as it takes on the role of the transistor between regular NP problems and $\mathsf{NP-hard}$ problems. In particular, no $\mathsf{NP-complete}$ problem is substantially harder than the others. Indeed, all of them are polynomially reducible to each other. The most difficult task was to find the very first $\mathsf{NP-complete}$ problem that needed to be reducible to all the other NP problems. This milestone was achieved by *Steven Cook* in 1971 [Coo71] who showed that the so-called *Satisfiability* problem is $\mathsf{NP-complete}$. One year later, the list of $\mathsf{NP-complete}$ problems increased considerably with Karp's 21 combinatorial problems [Kar72] and has since then experienced a tremendous expansion. Hereinafter, we give some more details on the Satisfiability problem and one of its variants, the 3-clause Satisfiability problem.

4.1 Satisfiability problem

A *Boolean expression* (or *logic formula*) φ is built from variables $v_1, \ldots, v_n$ that may take one of the two values TRUE (1) or FALSE (0), parenthesis, and the three operators:

- AND used for conjunction and denoted by $\wedge$,
- OR used for disjunction and denoted by $\vee$, and
- NOT used for negation and denoted by $\neg$.

A formula is said to be *satisfiable* if it can be made TRUE (1) by assigning appropriate logical values to its variables.

Definition 4.1. The *satisfiability problem* (SAT) asks to determine whether a given Boolean expression is satisfiable.

Despite its simple outline, the problem turns out to be extremely difficult. Indeed, if a formula is built from n variables, then there are 2^n possibilities to assess their Boolean values and, in the worst case, only a single assessment leads to TRUE (1). Nonetheless, it makes a good candidate for the pioneer $\mathsf{NP}-\mathsf{complete}$ problem.

Theorem 4.2 (Cook's Theorem [Coo71])**.** *The satisfiability problem is* $\mathsf{NP}-\mathsf{complete}$*.*

It is clear that a given witness can be verified in polynomial time using elementary Boolean logic. The difficult task is to show that any other NP problem can be reduced to it in polynomial time. We skip the details of this proof and refer the reader to Cook's original article [Coo71]. We highlight that the most significant contribution is that any other NP problem to which the Satisfiability problem can be reduced to in polynomial time becomes, through transitivity, automatically another $\mathsf{NP}-\mathsf{complete}$ problem.

4.2 Conjunctive normal form

To instantiate a more manageable version of the Satisfiability problem, we need some more insights on Boolean logic. A *literal* is either a variable v or its negation $\neg v$. A *clause* is a single literal or a disjunction of literals. A Boolean expression is in *conjunctive normal form* if it consists of a single clause or a conjunction of clauses. It is crucial to note that every Boolean expression can be transformed into an equivalent one in conjunctive normal form. Equivalence means that it contains the same variables and is satisfiable with a specific set of values $(v_1, \ldots, v_n)$ if and only if the original expression is satisfiable with this specific set of values $(v_1, \ldots, v_n)$. The drawback of this transformation is that the formula in conjunctive normal form may be exponentially larger than its original (in particular, it may not be a polynomial time transformation). The advantage of a conjunctive normal form is that the investigation of satisfiability becomes somewhat easier. Indeed, a Boolean expression in conjunctive normal form is satisfiable if and only if all its clauses are simultaneously satisfiable. Thereby, a simple contradiction between two clauses is enough to conclude unsatisfiability. Nonetheless, the remaining problem keeps being efficiently unsolved for now.

4.3 k-SAT

Considering Boolean formulas in conjunctive normal form allows one to systematically compare their solving methods. Thus, we may consider the Satisfiability problem with expressions in conjunctive normal form. Furthermore, to get even more control on the expressions we can restrict to formulas with clauses containing a fixed number of distinct literals only. This can be achieved by inserting, if necessary, auxiliary variables into the original expression. With this level of symmetry, we discover a new problem.

Definition 4.3. The *k-clause Satisfiability problem* (k-SAT) asks to determine whether a given Boolean expression in conjunctive normal form with exactly k distinct literals per clause is satisfiable.

Of course, if the clauses consist of isolated literals, satisfiability is trivial. Indeed, either the conjunction contains a variable and its negation and is so trivially unsatisfiable, or it does not contain both of them and is satisfiable. This proves that 1-SAT can be solved in linear time and is in P. The first nontrivial situation arises for clauses consisting of two distinct literals. Remarkably, the 2-SAT problem is still in P [Coo71] and even linear-time solving algorithms exist [APT79].

Theorem 4.4. *2-SAT problem is in* P.

The picture changes when considering larger clause sizes. An interesting by-product of Cook's proof for his theorem is that every satisfiability problem can be reduced to one with 3 literals per clause only. By transitivity, we thus deduce the following result [Coo71].

Theorem 4.5 (Cook's Theorem)**.** *3-SAT is* $\mathsf{NP-complete}$.

Chapter 5

Miscellaneous

This chapter prepares some auxiliary results that will be needed hereinafter. In Section 5.1 the number of solutions of a linear multivariate modular equation is computed. Section 5.2 approximates the volume of a higher dimensional sphere.

5.1 Solving linear multivariate congruences

First, we compute the number of solutions of multivariate equations modulo $q \in \mathbb{N}_{\geq 2}$.

Theorem 5.1. *Let $q \in \mathbb{N}_{\geq 2}$ and let $a_1, \ldots, a_n, b \in \mathbb{Z}$. Then, the congruence*

$$a_1 x_1 + \cdots + a_n x_n \equiv b \bmod q \tag{5.1}$$

has a solution $(x_1, \ldots, x_n) \in \mathbb{Z}^n$ if and only if $g = \gcd(a_1, \ldots, a_n, q)$ divides b. Furthermore, if Equation (5.1) has one solution, then it has gq^{n-1} distinct solutions in $\mathbb{Z}_q^n$ where $\mathbb{Z}_q := \mathbb{Z} \cap (-q/2, q/2]$.

Proof. Let $f : (\mathbb{Z}/q\mathbb{Z})^n \to (\mathbb{Z}/q\mathbb{Z})$ be the map defined by

$$f([x_1], \ldots, [x_n]) = \sum_{i=1}^{n} a_i [x_i]. \tag{5.2}$$

Then, Equation (5.1) has a solution if and only if $[b]$ belongs to the image of f. As $\mathrm{im}(f) = ([g])$ where $g = \gcd(a_1, \ldots, a_n, q)$ and $([g]) \subseteq \mathbb{Z}/q\mathbb{Z}$ denotes the ideal generated by $[g]$, we deduce the first part of the claim. Furthermore, in this case, the number of solutions of Equation (5.1) equals the number of elements in the kernel of f. By the *first isomorphism theorem* [Hum96, Theorem 8.13], $\mathrm{im}(f) \simeq (\mathbb{Z}/q\mathbb{Z})^n / \ker(f)$ and so

$$|\ker(f)| = \frac{|(\mathbb{Z}/q\mathbb{Z})^n|}{|\mathrm{im}(f)|} = \frac{q^n}{\frac{q}{g}} = gq^{n-1}$$

proving the second part of the claim. □

5.2 Volume of a sphere of radius R

Next, we develop an explicit upper bound for the number of lattice points inside a high-dimensional sphere. Our result is a slightly less precise, but explicit version of [MO90].

Theorem 5.2. *Let $m \in \mathbb{N}_{\geq 2}$. Then, the volume $V(\mathbb{S}_m(R))$ of the m-dimensional sphere of radius R defined by $\mathbb{S}_m(R) = \{x \in \mathbb{R}^m \mid \|x\|_2 = R\}$ is upper bounded by*

$$V(\mathbb{S}_m(R)) \leq \frac{1}{\sqrt{\pi m}} \left(\frac{2\pi e}{m}\right)^{\frac{m}{2}} R^m.$$

Proof

It is well known (e.g., [CS99, Section 2.C]) that the volume of the m-dimensional sphere is given by

$$V(\mathbb{S}_m(R)) = \frac{\pi^{\frac{m}{2}}}{\Gamma\left(\frac{m}{2}+1\right)} R^m, \tag{5.3}$$

where $\Gamma(z) = \int_0^{+\infty} x^{z-1} e^{-x} \; dx$ denotes the Gamma function. Thus, in order to prove our claim, it is sufficient to find a suitable lower bound for the Gamma function. This can be achieved by using some of its elementary properties (see [Abr74, Chapter 6] for all properties used hereinafter), as well as explicit versions of *Stirling's approximation* [Twe03] of the factorial function. We use the following theorem from [Rob55].

Theorem 5.3. *For all $n \in \mathbb{N}_{\geq 1}$, the factorial $n!$ is bounded by*

$$\sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n} e^{\frac{1}{12n+1}} < n! < \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n} e^{\frac{1}{12n}}.$$

Lemma 5.4. *Let $m \in \mathbb{N}_{\geq 2}$. Then,*

$$\Gamma\left(\frac{m}{2}+1\right) > \sqrt{\pi m} \left(\frac{m}{2e}\right)^{\frac{m}{2}}.$$

Proof. Assume first that $m = 2n$ for some $n \in \mathbb{N}_{\geq 1}$. Then:

$$\Gamma\left(\frac{m}{2}+1\right) = \Gamma(n+1) = n! \tag{5.4}$$

Using the lower bound of Theorem 5.3 yields

$$\Gamma\left(\frac{m}{2}+1\right) > \sqrt{2\pi}n^{n+\frac{1}{2}}e^{-n}e^{\frac{1}{12n+1}} = \sqrt{2\pi}\left(\frac{m}{2}\right)^{\frac{m+1}{2}}e^{-\frac{m}{2}}e^{\frac{1}{6m+1}}. \tag{5.5}$$

Rearranging the terms and observing that $e^{\frac{1}{6m+1}} > 1$ proves the claim.

Assume next that $m = 2n+1$ for some $n \in \mathbb{N}_{\geq 1}$. Then:

$$\Gamma\left(\frac{m}{2}+1\right) = \Gamma\left(n+1+\frac{1}{2}\right). \tag{5.6}$$

Using the *Legendre duplication formula* [MOS66, p.3], we deduce that

$$\Gamma\left(n+1+\frac{1}{2}\right) = 2^{1-2(n+1)}\sqrt{\pi}\frac{\Gamma(2(n+1))}{\Gamma(n+1)}. \tag{5.7}$$

As the entries of the Gamma functions are now positive integers, we may replace them by factorials:

$$\begin{aligned} 2^{1-2(n+1)}\sqrt{\pi}\frac{\Gamma(2(n+1))}{\Gamma(n+1)} &= 2^{1-2(n+1)}\sqrt{\pi}\frac{(2(n+1)-1)!}{n!} && (5.8)\\ &= 2^{-2(n+1)}\sqrt{\pi}\frac{(2(n+1))!}{(n+1)!}. && (5.9)\end{aligned}$$

Applying the lower bound of Theorem 5.3 to the factorial in the numerator and the upper bound to the factorial in the denominator, we deduce that

$$\begin{aligned} &2^{-2(n+1)}\sqrt{\pi}\frac{(2(n+1))!}{(n+1)!} && (5.10)\\ &> 2^{-2(n+1)}\sqrt{\pi}\frac{\sqrt{2\pi}(2(n+1))^{2(n+1)+\frac{1}{2}}e^{-2(n+1)}e^{\frac{1}{24(n+1)+1}}}{\sqrt{2\pi}(n+1)^{n+1+\frac{1}{2}}e^{-(n+1)}e^{\frac{1}{12n}}} && (5.11)\\ &= \sqrt{2\pi}\left(\frac{n+1}{e}\right)^{n+1}e^{-\frac{12n+25}{288n^2+300n}} && (5.12)\\ &= \sqrt{2\pi}\left(\frac{m+1}{2e}\right)^{\frac{m+1}{2}}e^{-\frac{6m+19}{72m^2+6m-78}}. && (5.13)\end{aligned}$$

This latter expression can be lower bounded by our claimed value:

$$\sqrt{2\pi}\left(\frac{m+1}{2e}\right)^{\frac{m+1}{2}} e^{-\frac{6m+19}{72m^2+6m-78}} > \sqrt{\pi m}\left(\frac{m}{2e}\right)^{\frac{m}{2}} \tag{5.14}$$

Indeed, rearranging the terms shows that the inequality is equivalent to

$$\sqrt{\frac{m+1}{me}}\left(\frac{m+1}{m}\right)^{\frac{m}{2}} > e^{\frac{6m+19}{72m^2+6m-78}}. \tag{5.15}$$

As only positive values are considered, squaring and applying the natural logarithm on both sides leads to the equivalent inequality

$$(m+1)\log\left(\frac{m+1}{m}\right) > 1 + \frac{6m+19}{36m^2+3m-39}. \tag{5.16}$$

Finally, Lemma 5.5 (below) proves correctness of Equation (5.16) for all odd integers $m \geq 3$, which in turn concludes the proof. □

Lemma 5.5. *The function $f : \mathbb{R}^+ \longrightarrow \mathbb{R}$ defined by*

$$f(x) = (x+1)\log\left(\frac{x+1}{x}\right) - 1 - \frac{6x+19}{36x^2+3x-39}$$

is positive for every positive odd integer strictly greater than 1.

Proof. It's first derivative is given by

$$f'(x) = \log\left(\frac{x+1}{x}\right) - \frac{432x^4 - 1389x^2 - 175x + 507}{3x(12x^2+x-13)^2} \tag{5.17}$$

and its second derivative by

$$\begin{aligned} &f''(x) \\ &= \frac{3456x^6 - 16848x^5 - 40140x^4 - 15911x^3 + 12013x^2 + 1521x - 6591}{3x^2(x+1)(12x^2+x-13)^3}. \end{aligned} \tag{5.18}$$

As the denominator in the second derivative is positive for all $x > 1$, the sign of the second derivative is determined by its numerator

$$3456x^6 - 16848x^5 - 40140x^4 - 15911x^3 + 12013x^2 + 1521x - 6591 \tag{5.19}$$

which has a single positive root $6 < x_0 < 7$. As $f''(x) > 0$ for all $x > x_0$, this implies that the first derivative f' is monotonically increasing on the interval $[7, +\infty[$. As $\lim_{x \to +\infty} f'(x) = 0$, we deduce that $f'(x) < 0$ for all $x \in [7, +\infty[$. Thus the considered function f is monotonically decreasing on the interval $[7, +\infty[$. As $\lim_{x \to +\infty} f(x) = 0$, we deduce that $f(x) > 0$ for all $x \in [7, +\infty[$. Furthermore, we computationally verify that $f(3) \approx 0.025$ and $f(5) \approx 0.038$. □

Using Lemma 5.4, we conclude that the Gamma function used in Equation (5.3) can be universally lower bounded by $\sqrt{\pi m}\left(\frac{m}{2e}\right)^{\frac{m}{2}}$, which is summarized in the next proposition.

Proposition 5.6. *For all $m \in \mathbb{N}_{\geq 2}$, $\Gamma(\frac{m}{2} + 1) > \sqrt{\pi m}\left(\frac{m}{2e}\right)^{\frac{m}{2}}$.*

Thus, we deduce that

$$V(\mathbb{S}_m(R)) = \frac{\pi^{\frac{m}{2}}}{\Gamma(\frac{m}{2} + 1)} R^m \tag{5.20}$$

$$\leq \frac{\pi^{\frac{m}{2}}}{\sqrt{\pi m}\left(\frac{m}{2e}\right)^{\frac{m}{2}}} R^m \tag{5.21}$$

$$= \frac{1}{\sqrt{\pi m}} \left(\frac{2\pi e}{m}\right)^{\frac{m}{2}} R^m \tag{5.22}$$

which proves Theorem 5.2. □

Chapter 6

Lattice preliminaries

Hereinafter, we give a short overview of some well-known lattice results. Our development is loosely based on the presentations in [Cas71, NV09, Gal12].

6.1 Lattices

The (integer row) *lattice* generated by the row vectors $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{Z}^m$ is the linear span

$$\Lambda = \mathcal{L}(\mathbf{v}_1, \dots, \mathbf{v}_n) := \left\{ \sum_{i=1}^{n} x_i \mathbf{v}_i \mid x_i \in \mathbb{Z} \; \forall i \in \{1, \dots, n\} \right\}. \tag{6.1}$$

If the vectors $\mathbf{v}_1, \dots, \mathbf{v}_n$ are linearly independent, they are called a basis of Λ. Similarly, a matrix $\mathbf{B}$ is called a *basis matrix* of Λ if the rows of $\mathbf{B}$ are linearly independent and Λ is generated by them (if the rows are not linearly independent, $\mathbf{B}$ is only called a *generating matrix*). It is well known that two bases $\mathbf{B}, \mathbf{B}'$ generate the same lattice if and only if there exists a unimodular matrix $\mathbf{U} \in GL(\mathbb{Z}, n)$ such that $\mathbf{B} = \mathbf{U}\mathbf{B}'$. The *dimension* of a lattice Λ is the dimension of its ambient space $\mathbb{Z}^m$. The *rank* of a lattice is its dimension as a $\mathbb{Q}$-span. A lattice is called *full-rank* if its rank is equal to its dimension.

6.2 Determinant

The *determinant* of a lattice Λ is defined by $\det(\Lambda) := \sqrt{\det(\mathbf{B}\mathbf{B}^T)}$ where $\mathbf{B}$ denotes any basis of Λ. Geometrically seen, the determinant of a lattice corresponds to the volume of its *fundamental parallelepiped* defined for a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ by

$$\mathcal{F}(\Lambda) := \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i \mid x_i \in [0,1) \; \forall i \in \{1, \ldots, n\} \right\}. \tag{6.2}$$

Furthermore, if $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ denotes the Gram-Schmidt orthogonalization of $\mathbf{b}_1, \ldots, \mathbf{b}_n$, then $\det(\Lambda) = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|_2$. Since $\|\mathbf{b}_i^*\|_2 \leq \|\mathbf{b}_i\|_2$ for all $i \in \{1, \ldots, n\}$, we deduce *Hadamard's inequality* [Had93]:

$$\det(\Lambda) \leq \prod_{i=1}^{n} \|\mathbf{b}_i\|_2. \tag{6.3}$$

6.3 Successive minima

For $i \in \{1, \ldots, n\}$, the i^{th} *successive minimum* of Λ denoted by $\lambda_i(\Lambda)$ is defined as the smallest radius $r > 0$ such that Λ contains at least i linearly independent vectors of length bounded by r. Symbolically,

$$\lambda_i(\Lambda) := \inf\{r \in \mathbb{R}_{>0} \mid \dim(\mathit{span}(\Lambda \cap B(0,r))) \geq i\} \tag{6.4}$$

where $B(0,r) := \{\mathbf{x} \in \mathbb{R}^m \mid \|\mathbf{x}\| \leq r\}$ denotes the closed ball of radius r around 0 for some norm $\|\cdot\|$ on $\mathcal{R}^m$. If not stated otherwise, we consider the Euclidean norm $\|\cdot\| = \|\cdot\|_2$. Naturally,

$$\lambda_1(\Lambda) \leq \cdots \leq \lambda_n(\Lambda). \tag{6.5}$$

The successive minima are achieved and the lattice vectors with norm $\lambda_i(\Lambda)$ are called the i-th shortest vectors, but they may not be unique.

6.4 Minkowski's theorems

For $n \in \mathbb{Z}_{\geq 1}$, the n-dimensional *Hermite constant* [Her50] is defined by

$$\gamma_n := \sup \left(\frac{\lambda_1(\Lambda)}{\det(\Lambda)^{\frac{1}{n}}} \right)^2 \tag{6.6}$$

where the supremum is taken over all rank n lattices. The exact value of γ_n is only known for $1 \leq n \leq 8$ [Wat66] and $n = 24$ [CK04]. For example, $\gamma_{24} = 4$. *Hermite's inequality* [Her50] yields $\gamma_n \leq \gamma_2^{n-1}$ with $\gamma_2 = \sqrt{\frac{4}{3}}$, and

[NV09] points out the linear bound $\gamma_n \leq 1 + \frac{n}{4}$, as well as the asymptotic behaviour

$$\frac{n}{2\pi e} + \frac{\log(\pi n)}{2\pi e} + o(1) \leq \gamma_n \leq \frac{1.744n}{2\pi e}(1 + o(1)). \tag{6.7}$$

Using *Minkowski's convex body theorem* [Min96] and a result by *Blichfeldt* [Bli14], one obtains *Minkowski's first theorem* [Min96] yielding an upper bound for the first lattice minimum:

$$\lambda_1(\Lambda) \leq \sqrt{\gamma_n} \det \Lambda^{\frac{1}{n}}. \tag{6.8}$$

Minkowski's second theorem [Min96] gives an upper bound for the geometric mean of the successive minima by stating that each $1 \leq i \leq n$:

$$\left(\prod_{j=1}^{i} \lambda_j(\Lambda) \right)^{1/i} \leq \sqrt{\frac{n}{2\pi e}} \det(\Lambda)^{1/n}. \tag{6.9}$$

6.5 Gaussian heuristic

The *Gaussian Heuristic* [Ajt06, GN08] predicts that for a "random" full-rank lattice of "large" dimension, we expect the shortest vector to not be much smaller than the value predicted by Minkowski's theorems. Symbolically,

$$\lambda_1(\Lambda) \simeq \sqrt{\frac{n}{2\pi e}} \det(\Lambda)^{1/n}. \tag{6.10}$$

The notion of "random" lattices can be made precise using the *Haar measure* [Ajt06], but we skip the details of this formalisation. We heuristically assume that for such lattices the lattice minima can be expected to be approximately of the same size, i.e., for all $i \in \{1, \ldots, n\}$

$$\lambda_i(\Lambda) \simeq \sqrt{\frac{n}{2\pi e}} \det(\Lambda)^{1/n}. \tag{6.11}$$

We assume that the same holds for lattices that are not full-rank.

6.6 Sublattices

We say that a lattice Λ' is a *sublattice* of a lattice Λ if $\Lambda' \subseteq \Lambda$. In this case, we call Λ a *superlattice* of Λ'. If both lattices have the same rank, we call

Λ' a *full-rank sublattice* of Λ. In this case, Λ' is also a subgroup of Λ and the group index $[\Lambda : \Lambda']$ is well-defined. Additionally, one can show that $[\Lambda : \Lambda'] = \frac{\det(\Lambda')}{\det(\Lambda)}$. Thus, $\det(\Lambda)$ divides $\det(\Lambda')$ and both lattices coincide if their determinants are equal.

6.7 Dual lattices

One may generalize integer lattices to *real lattices* by allowing real entries in the generating vectors (but still limiting to integer combinations only). For a real lattice Λ its *dual lattice* $\Lambda^\vee$ is defined by

$$\Lambda^\vee := \{\mathbf{w} \in \mathsf{span}_{\mathbb{R}}(\Lambda) \mid \langle \mathbf{v}, \mathbf{w} \rangle \in \mathbb{Z} \ \forall \mathbf{v} \in \Lambda\} \tag{6.12}$$

and consists of the set of linear functionals on Λ which take integer values on each lattice point of Λ. A basis $B^\vee$ for $\Lambda^\vee$ is obtained from a basis B of Λ by setting $B^\vee = B(B^T B)^{-1} \in \mathbb{R}^{n\times m}$. This basis relation yields that $(\Lambda^\vee)^\vee = \Lambda$ and $\det(\Lambda^\vee) = \frac{1}{\det(\Lambda)}$. *Banaszczyk's transference theorem* [Ban93] relates the successive minima of a lattice and its dual through the inequality:

$$1 \leq \lambda_k \lambda^\vee_{n-k+1} \leq n \quad \forall k \in \{1, \ldots, n\}. \tag{6.13}$$

6.8 Lattice reduction

Lattice reduction aims at computing so-called *reduced bases* with particular properties such as short and almost orthogonal basis vectors. We consider some particular lattice reductions that intend to approximate a shortest vector of a given lattice Λ.

6.8.1 Gauss-Lagrange reduction

For rank 2 lattices, the *Gauss-Lagrange* algorithm [Gal12, Section 17.1] can be used to reduce a given lattice basis. It essentially consists in a generalization of Euclid's greatest common divisors algorithm and manages to find a basis consisting of shortest vectors only. We note that for high rank lattices a basis consisting of shortest vectors only cannot be expected. Indeed, for $n \geq 5$, there exist so-called *non-standard* lattices that do not allow such a basis [FTW17]. Yet any lattice allows a basis containing a shortest vector [Ger08, Lemma 6.2].

6.8.2 LLL

The *Lenstra, Lenstra, and Lovász* (LLL) algorithm [LLL82] generalizes the Gauss-Lagrange algorithm. It has a polynomial runtime but manages only to find a basis containing an exponential approximation of a shortest vector. To be precise, let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a given basis of Λ and denote by $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ the associated *Gram-Schmidt* orthogonalization defined by $\mathbf{b}_1^* := \mathbf{b}_1$ and $\mathbf{b}_i^* := \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*$ for all $i \in \{2, \ldots, n\}$ where $\mu_{i,j} := \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle}$. Let $\frac{1}{4} < \delta < 1$. We say that $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is δ-*LLL reduced* if:

1. Size condition: $|\mu_{i,j}| \leq \frac{1}{2}$ for all $1 \leq j < i \leq n$.

2. Lovász condition: $\|\mathbf{b}_i^*\|_2^2 \geq (\delta - \mu_{i,i-1}^2)\|\mathbf{b}_{i-1}^*\|_2^2$ for all $2 \leq i \leq n$.

Traditionally, $\delta = \frac{3}{4}$, but in practice $\delta = 0.99$ is chosen. LLL reduced bases have many good properties such as

- $2^{\frac{1-i}{2}} \lambda_i(\Lambda) \leq \|\mathbf{b}_i\|_2 \leq 2^{\frac{n-1}{2}} \lambda_i(\Lambda)$ for all $i \in \{1, \ldots, n\}$, and

- $\det(\Lambda) = \prod_{i=1}^n \|\mathbf{b}_i\|_2 \leq \prod_{i=1}^n \|\mathbf{b}_i\|_2 \leq 2^{\frac{n(n-1}{2}} \det(\Lambda)$,

indicating that a reasonably good approximation of the shortest vectors is achieved but must be expected to include a blow-up term exponential in the lattice rank. The LLL-algorithm computes such a basis in time $O(n^5 m \log(B)^3)$ where B denotes the largest entry in absolute value of any basis vector $\mathbf{b}_1, \ldots, \mathbf{b}_n$. New variants of LLL such as [NS09, NSV11] achieve the same approximation with an improved time complexity. Nonetheless, hereinafter we make use of the original instantiation only.

6.8.3 BKZ

The *block Korkin-Zolotaref* (BKZ) algorithm [Sch87] generalizes the LLL algorithm by not only considering the relative sizes of two adjacent vectors in the basis (see the Lovász condition), but by comparing the size of β neighbouring vectors [GHGKN06]. This strongly improves the output quality of the reduction, but also increases its complexity. More precisely, for $\delta \geq 1$, we call a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ δ-*SVP reduced* [LN20] if $\|\mathbf{b}_1\|_2 = \delta\lambda_1(\Lambda)$. Given additionally a block size $2 \leq \beta \leq n$, we call the basis (δ, β)-*BKZ*-reduced [SE94] if it is size reduced and for every $i \in \{1, \ldots, n\}$, the basis $\mathbf{b}_i, \ldots, \mathbf{b}_{\min(i+\beta-1,n)}$ is δ-SVP reduced. We note that for $\delta = 1$ a shortest vector is given for each considered block of vectors. In this case, we have the following

$$\|\mathbf{b}_1\|_2 \leq \gamma_\beta^{\frac{n-1}{\beta-1}} \lambda_1(\Lambda) \tag{6.14}$$

where γ_β denotes the Hermite constant in dimension β [Sch87]. The BKZ algorithm finds such a basis in time exponential in the block size. Thus, despite the exactness of the shortest vector output, the BKZ algorithm suffers from a slow runtime. It is rather efficient for small block sizes $\beta \leq 30$ but degrades significantly beyond this bound. One solution is to use the recent improved BKZ 2.0 instantiation [CN11], which was particularly designed for large block sizes. However, its runtime is still exponential. For a fixed block size, stopping the algorithm after a polynomial number of rounds allows for a polynomial runtime, but decreases the precision of the output [LN20].

Chapter 7

Q-ary lattices

In this chapter, we give a probabilistic estimate on the size of the shortest vector of q-ary lattices. For a general overview of lattices and the corresponding notations we refer to Chapter 6. In Section 7.1, we follow [MR09] to define q-ary lattices. In Section 7.2, we give a probabilistic estimate of the shortest vector in a q-ary lattice in the infinity norm, and in Section 7.3 we do the same for the Euclidean norm.

7.1 Definition and properties

Definition 7.1. Let Λ be an integer row lattice. If $q\mathbb{Z}^m \subseteq \Lambda \subseteq \mathbb{Z}^m$ for some $q \in \mathbb{Z}_{\geq 2}$, then Λ is called a q-ary lattice.

By definition, every q-ary lattice has full rank $n = m$ as it contains the m linearly independent vectors $(q, 0, \ldots, 0), \ldots, (0, \ldots, 0, q)$. This also implies that the successive minima of a q-ary lattice are upper bounded by $\lambda_i(\Lambda) \leq q$ for all $i \in \{1, \ldots, m\}$. Given any matrix $\mathbf{A} \in \mathbb{Z}^{k \times m}$, we define the two special q-ary lattices:

$$\Lambda_q(\mathbf{A}) = \Big\{ \mathbf{y} \in \mathbb{Z}^m \mid \mathbf{y} \equiv \mathbf{x}\mathbf{A} \mod q \text{ for some } \mathbf{x} \in \mathbb{Z}^k \Big\} \tag{7.1}$$

representing the linear combinations of the rows of $\mathbf{A}$ modulo q, and

$$\Lambda_q^{\perp}(\mathbf{A}) = \big\{ \mathbf{y} \in \mathbb{Z}^m \mid \mathbf{y}\mathbf{A}^T \equiv 0 \mod q \big\} \tag{7.2}$$

representing systems of k linear homogeneous equations modulo q defined by the rows of $\mathbf{A}$. Interestingly, any q-ary lattice may be expressed as $\Lambda_q(\mathbf{A})$ or $\Lambda_q^{\perp}(\mathbf{A})$ for some matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times m}$ where $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ [MR09].

Furthermore, those two lattices are scaled duals. Indeed, $\Lambda_q^{\perp}(\mathbf{A}) = q\Lambda_q(\mathbf{A})^{\vee}$ and $\Lambda_q(\mathbf{A}) = q(\Lambda_q^{\perp}(\mathbf{A}))^{\vee}$. This yields that $\det(\Lambda_q(\mathbf{A}))\det(\Lambda_q^{\perp}(\mathbf{A})) = q^m$. Additionally, we obtain the following lemma.

Lemma 7.2. *Let q, m, k be fixed positive integers such that $m \geq k$, and let $\mathbf{A} \in \mathbb{Z}^{k\times m}$. Then,*

$$\det(\Lambda_q^{\perp}(\mathbf{A})) \leq q^k \textit{ and } \det(\Lambda_q(\mathbf{A})) \geq q^{m-k}$$

with equalities if and only if the rows of $\mathbf{A}$ are linearly independent modulo q.

Proof. As $\Lambda_q^{\perp}(\mathbf{A})$ is a full-rank sublattice of $\mathbb{Z}^m$, we derive $\det(\Lambda_q^{\perp}(\mathbf{A})) = [\mathbb{Z}^m : \Lambda_q^{\perp}(\mathbf{A})]$. Applying the *first isomorphism theorem* [Hum96, Theorem 8.13] to $f_{\mathbf{A}} : \mathbb{Z}^m \to \mathbb{Z}_q^k$ defined by $\mathbf{x} \mapsto f_{\mathbf{A}}(x) := \left[\mathbf{x}\mathbf{A}^T \mod q\right]$, we deduce that $|\mathbb{Z}^m/ker(f_{\mathbf{A}})| = |im(f_{\mathbf{A}})|$. As $ker(f_{\mathbf{A}}) = \Lambda_q^{\perp}(\mathbf{A})$ and $im(f_{\mathbf{A}}) \subseteq \mathbb{Z}_q^k$, we finally obtain

$$\det(\Lambda_q^{\perp}(\mathbf{A})) = [\mathbb{Z} : \Lambda_q^{\perp}(\mathbf{A})] = |\mathbb{Z}^m/ker(f_{\mathbf{A}})| = |im(f_{\mathbf{A}})| \leq q^k. \tag{7.3}$$

Furthermore, equality is achieved if and only if $im(f_{\mathbf{A}}) = \mathbb{Z}_q^k$ which in turn holds if and only if the rows of $\mathbf{A}$ are linearly independent modulo q. The claim for $\Lambda_q(\mathbf{A})$ follows from our previous observation that

$$\det(\Lambda_q(\mathbf{A}))\det(\Lambda_q^{\perp}(\mathbf{A})) = q^m. \tag{7.4}$$

□

Due to their special structure, q-ary lattices cannot be seen as random (as required for the Gaussian heuristic). Nonetheless, [AFG14] states that the Gaussian heuristic appears to hold exceedingly well for such lattices. If the rows of $\mathbf{A}$ are linearly independent modulo q, the previous lemma yields that the Gaussian heuristic for $\Lambda_q(\mathbf{A})$ would be $\lambda_1(\Lambda_q(\mathbf{A})) \simeq \sqrt{\frac{m}{2\pi e}}\, q^{(m-k)/n}$. In the next sections, we develop two precise probabilistic results on the size of the shortest vector of a q-ary lattice.

7.2 Shortest vector approximation - Infinity norm

In this section, we compute the probability of finding unusually short vectors of q-ary lattices in the infinity norm $\|\cdot\|_\infty$.

Theorem 7.3. *Let $q, m, k \in \mathbb{N}_{\geq 1}$ be fixed positive integers such that $m \geq k$. Then, the probability that for a uniformly at random chosen matrix $\mathbf{A} \in \mathbb{Z}_q^{k\times m}$, with $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$, the first lattice minimum of $\Lambda_q(\mathbf{A})$ satisfies*

$$\lambda_1^\infty(\Lambda_q(\mathbf{A})) \geq \frac{q^{\frac{m-k}{m}}}{4}$$

is at least $1 - 2^{-m}$. Here, $\lambda_1^\infty(\Lambda_q(\mathbf{A}))$ denotes the length of a shortest lattice vector of $\Lambda_q(\mathbf{A})$ in the infinity norm.

Remark 7.4. *The initial consideration of this probability was given in [Wen18, Lemma 2.17] and considered prime q only. Although the author classifies the result as well-known, we were not able to find it in the literature. Furthermore, the claimed proof contains a minor mistake as it only considers the first quadrant of the Euclidean space and neglects its symmetric copies around the origin. For a correct proof, we needed to slightly shrink the lower bound in the claimed probability.*

Proof. We want to analyse the opposite of the claimed probability, namely the probability that for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{k\times m}$ there exists a vector $\mathbf{x} \in \mathbb{Z}^k$ such that $\mathbf{xA} \equiv \mathbf{b} \mod q$ for some $\mathbf{b} \in \mathbb{Z}^m$ such that

$$0 < \|\mathbf{b}\|_\infty < \frac{q^{\frac{m-k}{m}}}{4} = B \tag{7.5}$$

As we are working modulo q, it suffices to consider $\mathbf{x} \in \mathbb{Z}_q^k$ and $\mathbf{b} \in \mathbb{Z}_q^m$. First, we note that the number of possible matrices $\mathbf{A} \in \mathbb{Z}_q^{k\times m}$ is q^{km}. Thus, it remains to determine the number of matrices that also satisfy the claimed property. We note that

$$\begin{aligned}
\mathcal{S} := &\left\{\mathbf{A} \in \mathbb{Z}_q^{k\times m} \mid \exists \mathbf{x} \in \mathbb{Z}_q^k, \exists \mathbf{b} \in \mathbb{Z}_q^m\colon\ \mathbf{xA} \equiv \mathbf{b} \mod q \ \wedge\ 0 < \|\mathbf{b}\|_\infty < B\right\} \\
\subseteq &\bigcup_{\substack{\mathbf{b}\in\mathbb{Z}_q^m \\ 0<\|\mathbf{b}\|_\infty<B}} \left\{\mathbf{A} \in \mathbb{Z}_q^{k\times m} \mid \exists \mathbf{x} \in \mathbb{Z}_q^k\colon\ \mathbf{xA} \equiv \mathbf{b} \mod q\right\} \qquad (7.6) \\
\subseteq &\bigcup_{\mathbf{x}\in\mathbb{Z}_q^k} \bigcup_{\substack{\mathbf{b}\in\mathbb{Z}_q^m \\ 0<\|\mathbf{b}\|_\infty<B}} \left\{\mathbf{A} \in \mathbb{Z}_q^{k\times m} \mid \mathbf{xA} \equiv \mathbf{b} \mod q\right\} \qquad (7.7)
\end{aligned}$$

The set $\left\{\mathbf{A} \in \mathbb{Z}_q^{k\times m} \mid \mathbf{xA} \equiv \mathbf{b} \mod q\right\}$ denotes all the solutions of the modular equation $\mathbf{xA} \equiv \mathbf{b} \mod q$ with fixed $\mathbf{x}$ and $\mathbf{b}$, and variable unknown $\mathbf{A}$. Each column of $\mathbf{A}$ and the corresponding entry of $\mathbf{b}$ give rise to an individual linear multivariate congruence. Indeed, letting $\mathbf{A} = (\mathbf{a}_1 \ldots \mathbf{a}_m)$ where

$\mathbf{a}_i$ denotes the i-th column of $\mathbf{A}$ and $\mathbf{b}_i$ denotes the i-th entry of the vector $\mathbf{b}$, we may consider the modular equation $\langle \mathbf{x}, \mathbf{a}_i \rangle \equiv \mathbf{b}_i \mod q$. By Theorem 5.1, we know that this equation can only have solutions if $g = \gcd(\mathbf{x}, q)$ divides $\mathbf{b}_i$. This holds only for a fraction $\frac{1}{g}$ of all values for $\mathbf{b}_i$ and, consequently, only for $\left(\frac{1}{g}\right)^m$ of all values for $\mathbf{b}$. In case of a suitable value $\mathbf{b}_i$, Theorem 5.1 implies that the equation has exactly gq^{k-1} solutions for $\mathbf{a}_i$, and consequently $(gq^{k-1})^m$ solutions for $\mathbf{A}$ if all entries of $\mathbf{b}$ allow a solution. Symbolically,

$$
\begin{aligned}
|\mathcal{S}| &\leq \sum_{\mathbf{x}\in\mathbb{Z}_q^k} \sum_{\substack{\mathbf{b}\in\mathbb{Z}_q^m \\ 0<\|\mathbf{b}\|_\infty<B}} \left|\left\{\mathbf{A} \in \mathbb{Z}_q^{k\times m} \mid \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q\right\}\right| && (7.8)\\
&= \sum_{\mathbf{x}\in\mathbb{Z}_q^k} \sum_{\substack{\mathbf{b}\in\mathbb{Z}_q^m \\ g|\gcd(\mathbf{b}) \\ 0<\|\mathbf{b}\|_\infty<B}} \left|\left\{\mathbf{A} \in \mathbb{Z}_q^{k\times m} \mid \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q\right\}\right| && (7.9)\\
&= \sum_{\mathbf{x}\in\mathbb{Z}_q^k} \sum_{\substack{\mathbf{b}\in\mathbb{Z}_q^m \\ g|\gcd(\mathbf{b}) \\ 0<\|\mathbf{b}\|_\infty<B}} (gq^{k-1})^m && (7.10)\\
&\leq \sum_{\mathbf{x}\in\mathbb{Z}_q^k} \left(\frac{2B}{g}\right)^m (gq^{k-1})^m && (7.11)\\
&= \sum_{\mathbf{x}\in\mathbb{Z}_q^k} (2Bq^{k-1})^m && (7.12)
\end{aligned}
$$

where the multiplicand $\left(\frac{2B}{g}\right)^m$ in Equation (7.11) stems from the fact that there are less than $2B$ distinct values for $\mathbf{b} \in \mathbb{Z}_q^m$ that satisfy $0 < \|\mathbf{b}\|_\infty < B$ (each entry of $\mathbf{b}$ ranges in $\{-B+1, \ldots, B-1\}$) and at most a fraction $\left(\frac{1}{g}\right)^m$ of such $\mathbf{b}$'s satisfy $g|\gcd(\mathbf{b})$. As the remaining sum runs over q^k values, we deduce that

$$
|\mathcal{S}| \leq q^k(2Bq^{k-1})^m = q^k\left(2\frac{q^{\frac{m-k}{m}}}{4}q^{k-1}\right)^m = \left(\frac{q^k}{2}\right)^m. \tag{7.13}
$$

Thus, the probability that for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{k\times m}$, there exists a vector $\mathbf{x} \in \mathbb{Z}^k$ and a vector $\mathbf{b} \in \mathbb{Z}^m$ such that $\mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q$ with

$0 < \|\mathbf{b}\|_\infty < \frac{q^{\frac{m-k}{m}}}{4}$ is

$$\mathbb{P}\left(\lambda_1^\infty(\Lambda_q(\mathbf{A})) < \frac{q^{\frac{m-k}{m}}}{4} \;\middle|\; \mathbf{A} \in \mathbb{Z}_q^{k\times m}\right) \leq \frac{\left(\frac{q^k}{2}\right)^m}{q^{km}} = 2^{-m}. \tag{7.14}$$

Considering the opposite event, and applying the complementary probability rule, finally leads to the desired conclusion:

$$\mathbb{P}\left(\lambda_1^\infty(\Lambda_q(\mathbf{A})) \geq \frac{q^{\frac{m-k}{m}}}{4} \;\middle|\; \mathbf{A} \in \mathbb{Z}_q^{k\times m}\right) \geq 1 - 2^{-m}. \tag{7.15}$$

□

7.3 Shortest vector approximation - Euclidean norm

In this section, we compute the probability of finding unusually short vectors of q-ary lattices in the Euclidean norm.

Theorem 7.5. *Let $q, m, k \in \mathbb{N}_{\geq 1}$ be fixed positive integers such that $m \geq k$. Then, the probability that for a uniformly at random chosen matrix $\mathbf{A} \in \mathbb{Z}_q^{k\times m}$, with $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$, the first lattice minimum of $\Lambda_q(\mathbf{A})$ satisfies*

$$\lambda_1(\Lambda_q(\mathbf{A})) \geq \min\left\{q, \sqrt{\frac{m}{8\pi e}} q^{\frac{m-k}{m}}\right\}$$

is at least $1 - \frac{1}{\sqrt{\pi m}} 2^{-m}$. Here $\lambda_1(\Lambda_q(\mathbf{A}))$ denotes the length of a shortest lattice vector of $\Lambda_q(\mathbf{A})$ in the Euclidean norm $\|\cdot\|_2$.

Remark 7.6. *The initial consideration of this probability was given in [Wen18, Lemma 2.18] and considered prime q only. We note that the term $\sqrt{\frac{m}{8\pi e}} q^{\frac{m-k}{m}}$ corresponds to half the Gaussian Heuristic of $\Lambda_q(\mathbf{A})$ if the rows of $\mathbf{A}$ are linearly independent.*

Proof. By definition, the q-ary lattice $\Lambda_q(\mathbf{A})$ contains the m linearly independent vectors $(q, 0, \ldots, 0), \ldots, (0, \ldots, 0, q)$. Thus, the first lattice minimum is trivially upper bounded by q and so it is sufficient to compute the probability for $\lambda_1(\Lambda_q(\mathbf{A})) \geq B$ where

$$B = \sqrt{\frac{m}{8\pi e}} q^{\frac{m-k}{m}} < q \tag{7.16}$$

Actually, we analyse the opposite of the claimed probability, namely the probability that for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{k \times m}$ there exist vectors $\mathbf{x} \in \mathbb{Z}^k$ and $\mathbf{b} \in \mathbb{Z}^m$ such that $\mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q$ with $0 < \|\mathbf{b}\|_2 < B$. As we work modulo q, it suffices to consider $\mathbf{x} \in \mathbb{Z}_q^k$ and $\mathbf{b} \in \mathbb{Z}_q^m$. We remark that the number of possible matrices $\mathbf{A} \in \mathbb{Z}_q^{k \times m}$ is q^{km}. Thus, it remains to determine the number of matrices that also satisfy the claimed property. We note that

$$\begin{aligned}
\mathcal{S} := &\left\{\mathbf{A} \in \mathbb{Z}_q^{k \times m} \;\middle|\; \exists \mathbf{x} \in \mathbb{Z}_q^k, \exists \mathbf{b} \in \mathbb{Z}_q^m \colon\; \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q \;\wedge\; 0 < \|\mathbf{b}\|_2 < B\right\} \\
\subseteq &\bigcup_{\substack{\mathbf{b} \in \mathbb{Z}_q^m \\ 0 < \|\mathbf{b}\|_2 < B}} \left\{\mathbf{A} \in \mathbb{Z}_q^{k \times m} \;\middle|\; \exists \mathbf{x} \in \mathbb{Z}_q^k \colon\; \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q\right\} \qquad (7.17) \\
\subseteq &\bigcup_{\mathbf{x} \in \mathbb{Z}_q^k} \bigcup_{\substack{\mathbf{b} \in \mathbb{Z}_q^m \\ 0 < \|\mathbf{b}\|_2 < B}} \left\{\mathbf{A} \in \mathbb{Z}_q^{k \times m} \;\middle|\; \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q\right\} \qquad (7.18)
\end{aligned}$$

The set $\left\{\mathbf{A} \in \mathbb{Z}_q^{k \times m} \mid \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q\right\}$ denotes all the solutions of the modular equation $\mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q$ with fixed $\mathbf{x}$ and $\mathbf{b}$ and variable unknown $\mathbf{A}$. Each column of $\mathbf{A}$ and the corresponding entry of $\mathbf{b}$ give rise to an individual linear multivariate congruence. Indeed, letting $\mathbf{A} = (\mathbf{a}_1 \ldots \mathbf{a}_m)$ where $\mathbf{a}_i$ denotes the i-th column of $\mathbf{A}$ and $\mathbf{b}_i$ denotes the i-th entry of the vector $\mathbf{b}$, we may consider the modular equation $\langle \mathbf{x}, \mathbf{a}_i \rangle \equiv \mathbf{b}_i \mod q$. By Theorem 5.1, we know that these equations can only have solutions if the divisor condition $g = \gcd(\mathbf{x}, q) | \mathbf{b}_i$ is satisfied. This holds only for a fraction $\frac{1}{g}$ of all values for $\mathbf{b}_i$ and consequently only for $\left(\frac{1}{g}\right)^m$ of all values for $\mathbf{b}$. In case of a suitable value $\mathbf{b}_i$, Theorem 5.1 implies that the equation has exactly gq^{k-1} solutions for $\mathbf{a}_i$, and consequently $(gq^{k-1})^m$ solutions for $\mathbf{A}$ if all entries of $\mathbf{b}$ allow a solution. Symbolically,

$$\begin{aligned}
|\mathcal{S}| &\leq \sum_{\mathbf{x} \in \mathbb{Z}_q^k} \sum_{\substack{\mathbf{b} \in \mathbb{Z}_q^m \\ 0 < \|\mathbf{b}\|_2 < B}} \left|\left\{\mathbf{A} \in \mathbb{Z}_q^{k \times m} \;\middle|\; \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q\right\}\right| \qquad (7.19) \\
&= \sum_{\mathbf{x} \in \mathbb{Z}_q^k} \sum_{\substack{\mathbf{b} \in \mathbb{Z}_q^m \\ g | \gcd(\mathbf{b}) \\ 0 < \|\mathbf{b}\|_2 < B}} \left|\left\{\mathbf{A} \in \mathbb{Z}_q^{k \times m} \;\middle|\; \mathbf{x}\mathbf{A} \equiv \mathbf{b} \mod q\right\}\right| \qquad (7.20) \\
&= \sum_{\mathbf{x} \in \mathbb{Z}_q^k} \sum_{\substack{\mathbf{b} \in \mathbb{Z}_q^m \\ g | \gcd(\mathbf{b}) \\ 0 < \|\mathbf{b}\|_2 < B}} (gq^{k-1})^m \qquad (7.21)
\end{aligned}$$

$$\leq \sum_{\mathbf{x}\in\mathbb{Z}_q^k} \frac{\frac{1}{\sqrt{\pi m}}\left(\frac{2\pi e}{m}\right)^{\frac{m}{2}} B^m}{g^m} (gq^{k-1})^m \tag{7.22}$$

$$= \sum_{\mathbf{x}\in\mathbb{Z}_q^k} \frac{1}{\sqrt{\pi m}}\left(\frac{2\pi e}{m}\right)^{\frac{m}{2}} B^m q^{mk-m} \tag{7.23}$$

where the multiplicand $\frac{\frac{1}{\sqrt{\pi m}}\left(\frac{2\pi e}{m}\right)^{\frac{m}{2}} B^m}{g^m}$ in Equation (7.22) stems from the fact that there are less than $\frac{1}{\sqrt{\pi m}}\left(\frac{2\pi e}{m}\right)^{\frac{m}{2}} B^m$ distinct values for $\mathbf{b} \in \mathbb{Z}_q^m$ that satisfy $0 < \|\mathbf{b}\|_2 < B$ (see Theorem 5.2) and at most a fraction $\left(\frac{1}{g}\right)^m$ of such $\mathbf{b}$'s satisfy $g|\gcd(\mathbf{b})$. As the remaining sum runs over q^k values, we deduce that

$$|\mathcal{S}| \leq q^k \frac{1}{\sqrt{\pi m}}\left(\frac{2\pi e}{m}\right)^{\frac{m}{2}} B^m q^{mk-m} = \frac{1}{\sqrt{\pi m}}\left(\frac{q^k}{2}\right)^m. \tag{7.24}$$

Thus, the probability that for a random matrix $\mathbf{A} \in \mathbb{Z}_q^{k\times m}$, there exist vectors $\mathbf{x} \in \mathbb{Z}^k$ and $\mathbf{b} \in \mathbb{Z}^m$ such that $\mathbf{xA} \equiv \mathbf{b} \bmod q$ with $0 < \|\mathbf{b}\|_2 < \sqrt{\frac{m}{8\pi e}} q^{\frac{m-k}{m}} = B$ is

$$\mathbb{P}\left(\lambda_1^2(\Lambda_q(\mathbf{A})) < \frac{\sqrt{m}q^{\frac{m-k}{m}}}{2\sqrt{2\pi e}} \;\middle|\; \mathbf{A} \in \mathbb{Z}_q^{k\times m}\right) \leq \frac{\frac{1}{\sqrt{\pi m}}\left(\frac{q^k}{2}\right)^m}{q^{km}} = \frac{2^{-m}}{\sqrt{\pi m}}. \tag{7.25}$$

Considering now the opposite event and applying the complementary probability rule finally leads to the desired conclusion:

$$\mathbb{P}\left(\lambda_1^2(\Lambda_q(\mathbf{A})) \geq \frac{\sqrt{m}q^{\frac{m-k}{m}}}{2\sqrt{2\pi e}} \;\middle|\; \mathbf{A} \in \mathbb{Z}_q^{k\times m}\right) \geq 1 - \frac{1}{\sqrt{\pi m}} 2^{-m}. \tag{7.26}$$

□

Bibliography I

[Abr74] Milton Abramowitz. *Handbook of Mathematical Functions, with Formulas, Graphs, and Mathematical Tables*. Dover Publications, Inc., USA, 1974.

[AFG14] Martin Albrecht, Robert Fitzpatrick, and Florian Göpfert. On the efficacy of solving LWE by reduction to unique-SVP. *Information Security and Cryptology – ICISC 2013*, pages 293–310, 2014.

[Ajt06] Miklós Ajtai. Generating random lattices according to the invariant distribution. Draft, March 2006.

[APT79] Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[Ban93] Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(4):625–636, 1993.

[Bli14] Hans F. Blichfeldt. A new principle in the geometry of numbers, with some applications. *Trans. Amer. Math. Soc.*, 15(3):227–235, 1914.

[Cas71] John W. S. Cassels. *An Introduction to the Geometry of Numbers.* Die Grundlehren der mathematischen Wissenschaften, Band 99. Springer-Verlag, Berlin-New York, 1971.

[CK04] Henry Cohn and Abhinav Kumar. The densest lattice in twenty-four dimensions. *Electron. Res. Announc. Amer. Math. Soc.*, 10:58–67, 2004.

[CMI00] Clay Mathematics Institute CMI. P vs NP problem. http://www.claymath.org/millennium-problems/p-vs-np-problem, 2000. Last accessed 26.04.2022.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 1–20, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery.

[Coo00] Stephen Cook. The P versus NP problem. In *Clay Mathematical Institute; The Millennium Prize Problem*, 2000.

[CS99] John H. Conway and Neil J. A. Sloane. *Voronoi Cells of Lattices and Quantization Errors*, pages 451–477. Springer New York, New York, NY, 1999.

[FTW17] Rongquan Feng, Longke Tang, and Kun Wang. On the standard lattices. arXiv Number 1703.08765, 2017.

[Gal12] Steven D. Galbraith. *Mathematics of Public Key Cryptography.* Cambridge University Press, Cambridge, 2012.

[Ger08] Larry J. Gerstein. *Basic Quadratic Forms*, volume 90 of *Graduate Studies in Mathematics.* American Mathematical Society, Providence, RI, 2008.

[GHGKN06] Nicolas Gama, Nick Howgrave-Graham, Henrik Koy, and Phong Q. Nguyen. Rankin's constant and blockwise lattice reduction. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 112–130, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 31–51. Springer, 2008.

[Had93] Jacques Hadamard. Résolution d'une question relative aux déterminants. *Bull. des Sciences Math.*, 2(17):240–246, 1893.

[Her50] Charles Hermite. Extraits de lettres de M. Ch. Hermite à M. Jacobi sur différents objects de la théorie des nombres. (Continuation). *J. Reine Angew. Math.*, 40:279–315, 1850.

[HKO01] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*, volume 26. 2001.

[Hum96] John F. Humphreys. *A Course in Group Theory.* Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1996.

[Kar72] Richard Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, volume 40, pages 85–103, 01 1972.

[Lee90] Jan V. Leeuwen. *Handbook of Theoretical Computer Science: Algorithms and Complexity.* MIT Press, Cambridge, MA, USA, 1990.

[LLL82] Arjen Lenstra, H. Lenstra, and Lovász László. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261, 12 1982.

[LN20] Jianwei Li and Phong Q. Nguyen. A complete analysis of the BKZ lattice reduction algorithm. Cryptology ePrint Archive, Report 2020/1237, 2020. https://ia.cr/2020/1237.

[Min96] Hermann Minkowski. *Geometrie der Zahlen.* Teubner, Leibzig, 1896.

[MO90] James E. Mazo and Andrew M. Odlyzko. Lattice points in high-dimensional spheres. *Monatsh. Math.*, 110(1):47–61, 1990.

[MOS66] Wilhelm Magnus, Fritz Oberhettinger, and Raj Pal Soni. *Formulas and Theorems for the Special Functions of Mathematical Physics.* Die Grundlehren der mathematischen Wissenschaften, Band 52. Springer-Verlag New York, Inc., New York, enlarged edition, 1966.

[MR09] Daniele Micciancio and Oded Regev. *Lattice-based Cryptography*, pages 147–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[NS09] Phong Q. Nguyen and Damien Stehlé. An LLL algorithm with quadratic complexity. *SIAM Journal on Computing*, 39(3):874–903, 2009.

[NSV11] Andrew Novocin, Damien Stehlé, and Gilles Villard. An LLL-reduction algorithm with quasi-linear time complexity: Extended abstract. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 403–412, New York, NY, USA, 2011. Association for Computing Machinery.

[NV09] Phong Q. Nguyen and Brigitte Valle. *The LLL Algorithm: Survey and Applications.* Springer Publishing Company, Incorporated, 1st edition, 2009.

[Rob55] Herbert Robbins. A remark on Stirling's formula. *The American Mathematical Monthly*, 62(1):26–29, 1955.

[Sch87] Claus P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, 1987.

[SE94] Claus P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66(2):181–199, sep 1994.

[Shy13] John J. Shynk. *Probability, Random Variables, and Random Processes.* Wiley, 1st edition, 2013.

[Twe03] Ian Tweddle. *James Stirling's Methodus Differentialis.* Sources and Studies in the History of Mathematics and Physical Sciences. Springer-Verlag London, Ltd., London, 2003. An annotated translation of Stirling's text.

[Wat66] George L. Watson. On the minimum of a positive quadratic form in $n\,(\leq 8)$ variables. Verification of Blichfeldt's calculations. *Proc. Cambridge Philos. Soc.*, 62:719, 1966.

[Wen18] Weiqiang Wen. *Contributions to the hardness foundations of lattice-based cryptography.* PhD thesis, Université de Lyon, Computational Complexity, 2018. HAL.

Part II

Simultaneous Chinese Remaindering

Act II: The count

Little did Jay know about the work and rigour of a coin collector. Coins should be treated with care. Some are allowed to be cleaned, others not. New coins must be analysed with microscopic precision and the historical background needs to be worked out. "Mastering the right machinery was a difficult task, but finally, I'm ready!" told Jay his pa.

That weekend, the annual "Big Count" was planned – an inventory of all coins, tokens, and paper money in the family collection. Jay was proud to have been assigned the counting of pennies but has been surprised by the sheer quantity that pops put on the kitchen table. The new ones are made of copper, and the old ones of nickel, zinc, or bronze. At least two wheelbarrows of those one-cent U.S. coins perfectly blended in with the brown oak wood table.

Counting was arduous. Jay tried first to count by stacks of fives, but soon realised that there is not enough space to store them. Then he tried to pile twenty at a time, but shortly after the staples collapsed, ruining his count again. After watching the hilarious performance of his son, Jay's pa gave him a hint: "You don't need to form individual piles. Simply choose a count value, such as three, five, or seven, and always put this number of coins aside on a big heap. The only important quantity is the number of remaining pennies at the end. If you count those remainders for sufficiently many count values, we can combine the remainder information and find out the total number of coins."

Astonished that such an easy way of counting exists, Jay started over by applying his pa's technique, and soon he was done with the task. As it was already late, he went to bed. The next morning, he was terrified as he spotted Mrs Skizzles on his notebook taking a bath. When he hushed her away it was too late, some numbers were already unreadable such that for a single count value several remainders were possible. "Maybe pa can still figure out the total number of pennies, or at least an approximation of it", hoped Jay.

Abstract II

The classical Chinese Remainder Problem asks to find all integer solutions to a given system of congruences where each congruence is defined by one modulus and one remainder. If solutions exist, they are completely defined by the minimal positive solution and the lowest common multiple of the considered moduli. It is well known that there are efficient algorithms to find this minimal positive solution and so the Chinese Remainder Problem can be solved in polynomial time. Hereinafter, we consider a direct generalization of the Chinese Remainder Problem where not only a single remainder is given per modulus but each modulus is accompanied by a non-empty set of remainders. We call this new problem the Simultaneous Chinese Remainder Problem. The solutions of a problem instance are completely defined by a set of minimal positive solutions upper bounded by the lowest common multiple of the considered moduli. The size of the set of minimal positive solutions grows exponentially in the remainder set sizes and so any solving method requires exponential time. Through a direct reduction from the 3-SAT problem, we prove that already deciding whether a solution exists is NP-complete. Similarly, we show that if the existence of solutions is guaranteed, then deciding whether a solution of a particular size exists is still NP-complete. We deepen this result by studying the minimal solution of a Simultaneous Chinese Remainder Problem instance. First, we develop some rough upper bounds and then we concentrate on concrete solving methods. Naturally, no polynomial time algorithm can be expected to find the minimal solution, yet some insights are gained. Subsequently, we present some experimental results and discuss corresponding heuristics. We finish our study with a list of open questions.

Contents

Act II **53**

Abstract II **54**

Contents II **55**

8 Chinese Remaindering **59**

8.1 The Chinese Remainder Problem 59

8.2 Complexity of Chinese Remaindering 60

8.2.1 Corresponding decision problems 60

8.2.2 The case of two congruences 61

8.2.3 The multi-congruence case 62

8.3 Chinese Remainder Theorem 63

8.4 Particular solving methods 63

8.4.1 Systematic search 64

8.4.2 Textbook CRT 64

8.4.3 Garner's algorithm 64

8.5 The power of Chinese Remaindering 65

8.6 Chinese Remaindering and Polynomial Interpolation 66

8.6.1 Lagrange 66

8.6.2 Newton 67

9 Simultaneous Chinese Remaindering **69**

9.1 The Simultaneous Chinese Remainder Problem 69

9.2 Complexity of Simultaneous Chinese Remaindering 70

9.3 Recursive Garner 71

10 Two Simultaneous Chinese Remainder Decision Problems **73**

10.1 Existential Simultaneous Chinese Remainder Problem 73

10.1.1 Proof intuition 74

10.1.2 Proof structure 74
10.1.3 A. Membership in NP 74
10.1.4 B. Problem construction 75
10.1.5 C. Solution matching 77
10.2 Bounded Simultaneous Chinese Remainder Problem 78
10.2.1 Proof intuition 78
10.2.2 Comment 79
10.2.3 Proof structure 79
10.2.4 A. Membership in NP 79
10.2.5 B. Problem construction 80
10.2.6 Bound analysis 82
10.2.7 Some observations 83
10.2.8 C. Solution matching 86
10.2.9 The hardness of different instances 87
10.2.10 A generalization 87

11 Simultaneous Chinese Remainder Problem variants 91
11.1 The Minimal Simultaneous Chinese Remainder Problem . . . 91
11.2 Other related problems 92
11.3 Problems in the literature 94
11.4 Applications 95
11.4.1 Factoring with extra information 95
11.4.2 Point counting on elliptic curves 96

12 An upper bound for the minimal solution 97
12.1 Intuition 97
12.2 Maximal number of solutions inside a given interval 98
12.3 Elementary upper bound 99
12.4 Optimal ordering 101
12.5 Limitations 102

13 Deterministic solving method 103
13.1 The trivial case: k=1 103
13.2 The first non-trivial case: k=2 103
13.2.1 A solving algorithm 104
13.2.2 Analysis 104
13.3 The general case 105

14 Family of minimizing functions **107**
14.1 Declaring a new modulus and auxiliary coefficients 107
14.2 Constructing new functions minimizing in an extremum . . . 108
14.3 Intermediate sizes 112
14.4 A family of functions minimized in an extremum 113

15 Lattice solving methods **115**
15.1 Coppersmith's method 115
15.1.1 The approach 116
15.1.2 Comments 117
15.2 Bleichenbacher-Nguyen method 118
15.2.1 First approach 118
15.2.2 Remarks 119
15.2.3 Second approach 120

16 Reformulations **125**
16.1 A sieving problem 125
16.1.1 Intuition 125
16.1.2 Auxiliary notations 126
16.1.3 Elementary sieve results 127
16.1.4 Brun's sieve 129
16.2 A subset sum problem 130
16.2.1 The multiple-choice knapsack problem and its subset sum equivalent 131
16.2.2 Problem construction 131
16.2.3 Remarkable results 132

17 An illustrative example **133**

18 Experiments and heuristics **135**
18.1 Cumulative distribution 135
18.2 Fixed remainder set size 138
18.2.1 Closed form formula 138
18.2.2 Recursive counting function 139
18.2.3 A constructive approach 141
18.2.4 Improved counting method 144
18.2.5 Particular observations 146
18.3 Simulating the distributions 148
18.4 The maximal minimal solution 150
18.5 Remainder sets of size two 152

18.6 Moduli with common divisors 154

19 Open Questions **155**

Bibliography II **157**

Chapter 8

Chinese Remaindering

In the third century book *Sunzi Suanjing* [Kat93, Section 7.5.1], the following puzzle was raised:

> *"There are certain things whose number is unknown. If we count them by threes, we have two leftovers; by fives, we have three leftovers and by sevens, two are left over. How many things are there?"*

Today, we know that the solution to this riddle is 23 or more precisely $23+k\cdot 105$ for any $k \in \mathbb{N}$. This chapter is devoted to the general formalization of this problem and a short review of its solving methods.

8.1 The Chinese Remainder Problem

The *Chinese Remainder Problem* has first been announced and solved in all generality by *Qin Jiushao* in his *Mathematical Treatise in Nine Sections* [Lib73]. Subsequently, it was put into its modern form by *Gauss* [Gau95] using the notion of congruences.

Definition 8.1 (Chinese Remainder Problem (CRP))**.** Let $m_1, \ldots, m_k \in \mathbb{N}$ be positive integers, and, for all $i \in \{1, \ldots, k\}$, let $r_i \in \{0, \ldots, m_i - 1\}$ denote a remainder modulo m_i. Find, if it exists, $x \in \mathcal{S}_M$, where $\mathcal{S}_M$ denotes a set of $M = \operatorname{lcm}(m_1, \ldots, m_k)$ consecutive integers, such that

$$\begin{cases} x & \equiv & r_1 & \mod m_1, \\ & \vdots & & \vdots \\ x & \equiv & r_k & \mod m_k. \end{cases}$$

We refer to this problem by $\mathsf{CRP}((m_1, r_1), \ldots, (m_k, r_k), \mathcal{S}_M)$ and to x, if it exists, as the *primitive solution*. Any other integer $x' \notin \mathcal{S}_M$ satisfying all the congruences is called *non-primitive solution*.

We note that the given value restrictions do not have an impact on the generality of the problem. Indeed, any $r'_i \in \mathbb{Z}$ is congruent to some $r_i \in \{0, \ldots, m_i - 1\}$ modulo m_i such that the choice of the remainder representatives does not matter. Furthermore, if there exists a solution $x \in \mathbb{Z}$, then $y \in \mathbb{Z}$ is a solution if and only if $y = x + k \operatorname{lcm}(m_1, \ldots, m_k)$ for some $k \in \mathbb{Z}$. In particular, in this case, there is a unique $\chi \in \mathcal{S}_M$ that satisfies all congruences.

Remark 8.2. *The set $\mathcal{S}_M$ in Definition 8.1 allows us to switch between different solution representatives, such as the canonical representatives $\{0, 1, \ldots, M\}$ and the symmetric representatives $\mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$ where $M = \operatorname{lcm}(m_1, \ldots, m_k)$.*

In this chapter, we consider $\mathcal{S}_M = \{0, \ldots, \operatorname{lcm}(m_1, \ldots, m_k) - 1\}$ only and so we use the simplified notation $\mathsf{CRP}((m_1, r_1), \ldots, (m_k, r_k))$.

8.2 Complexity of Chinese Remaindering

In this section, we quickly investigate the complexity of Chinese Remaindering. We refer to Chapter 3 for our complexity-theoretic framework.

8.2.1 Corresponding decision problems

The traditional Chinese Remainder Problem consists in a search problem requiring a non-binary integer solution. For an easy comparison in Chapter 10, we raise the following two decision problems.

Definition 8.3. The *Existential* Chinese Remainder Problem asks to determine whether a given Chinese Remainder Problem $\mathsf{CRP}((m_1, r_1), \ldots, (m_k, r_k))$ has a solution.

Definition 8.4. The *Bounded* Chinese Remainder Problem asks us to determine whether a given Chinese Remainder Problem $\mathsf{CRP}((m_1, r_1), \ldots, (m_k, r_k))$ has a solution $x < B$ for some predefined $B \in \mathcal{S}_M$.

We prove that the initial search problem and the invoked decision problems can be solved at essentially the same cost. In particular, we devise a polynomial-time solving algorithm that solves all of them simultaneously.

8.2.2 The case of two congruences

When considering two congruences only, the Chinese Remainder Problem takes the form

$$\begin{cases} x \equiv r_1 \mod m_1, \\ x \equiv r_2 \mod m_2. \end{cases} \tag{8.1}$$

Rewriting these congruences as Diophantine equations leads to solving

$$\begin{cases} x = r_1 + k_1 m_1, \\ x = r_2 + k_2 m_2. \end{cases} \tag{8.2}$$

for some $k_1, k_2 \in \mathbb{Z}$. Put differently, we need to solve the bivariate linear Diophantine equation

$$r_1 + k_1 m_1 = r_2 + k_2 m_2. \tag{8.3}$$

Restructuring both sides of the equation leads to

$$k_1 m_1 - k_2 m_2 = r_2 - r_1. \tag{8.4}$$

Recognizing the special form of this equation, we note that it can be solved by the *Extended Euclidean Algorithm.* Indeed, on input (m_1, m_2), the Extended Euclidean Algorithm returns a triple (g, t_1, t_2) such that $g = \gcd(m_1, m_2)$ and $g = t_1 m_1 + t_2 m_2$. Thus, Equation (8.4) has a solution if and only if g divides $r_2 - r_1$, in which case $k_1 = t_1 \frac{r_2 - r_1}{g}$ and $k_2 = -t_2 \frac{r_2 - r_1}{g}$. The primitive solution x to the initial Chinese Remainder Problem is then obtained as

$$x = \left[\!\!\left[r_1 + t_1 \frac{r_2 - r_1}{g} m_1 \mod \frac{m_1 m_2}{g} \right]\!\!\right] \tag{8.5}$$

where $[\![a \mod b]\!]$ returns the smallest non-negative remainder of a modulo b. To limit the size of the intermediate values, all of these computations can be carried out modulo $\frac{m_1 m_2}{g}$. A slightly improved computation is obtained by

$$x = \left[\!\!\left[r_1 + \left[\!\!\left[t_1 \frac{r_2 - r_1}{g} \mod \frac{m_2}{g} \right]\!\!\right] m_1 \mod \frac{m_1 m_2}{g} \right]\!\!\right]. \tag{8.6}$$

The time complexity of the recombination procedure is dominated by the Extended Euclidean Algorithm. If $\max\{m_1, m_2\} < 2^n$, then the usual textbook Extended Euclidean Algorithm runs in time $O(n^2)$ and the *Schönhage* controlled Euclidean descent performs the same task in time $O(k \log^2(n))$ [Mö08]. The resulting triple (g, t_1, t_2) has no entry larger than the initial moduli [FH96]. So, the multiplication $t_1 \frac{r_2 - r_1}{g}$ can be performed in time

$O(n^2)$ with the usual textbook multiplication and in time $O(k \log(n))$ with the *Harvey-Hoeven* algorithm [HvdH21]. The subsequent modular reduction modulo $\frac{m_2}{g}$ takes place in time $O(n^2)$ as at most n subtractions of an n-bit integer are required. Under *Newton-Raphson* [GG13, Theorem 9.8], the required divisions are as efficient as the considered multiplication. Similarly, the second multiplication with m_1 is performed in time $O(n^2)$ and the final modular reduction modulo $\frac{m_1 m_2}{g}$ takes time $O(n^2)$ as at most a single reduction of an n^2 bit integer is needed. Thus, the overall time complexity is $O(n^2)$ and so quadratic in the input size.

8.2.3 The multi-congruence case

The simple two congruence case can be generalised to k congruences through a recursive process. More precisely, we simplify the system

$$\begin{cases} x & \equiv & r_1 & \mod m_1, \\ & \vdots & & \vdots \\ x & \equiv & r_k & \mod m_k, \end{cases} \tag{8.7}$$

by treating one congruence at a time. First, we apply the solving method from Section 8.2.2 to the first two congruences modulo m_1 and m_2. Thereby, we deduce the congruence $x \equiv r_{1,2} \mod \left(\frac{m_1 m_2}{g}\right)$. Then, we apply the same method to this new congruence and the congruence modulo m_3. Inductively, we repeat the procedure until we combined all the congruences to a single one. As the time complexity of the two congruence case is determined by the length of the considered moduli, we observe that the application becomes heavier with each iteration. Indeed, assuming that $\max(m_1, \dots, m_k) < 2^n$, the first iteration takes place in time $O(n^2)$. The subsequent iteration includes one modulus that is smaller than 2^n and one that is only smaller than 2^{2n}. Thus, the time complexity grows to $O(4n^2)$. The same trend is maintained until the last iteration, which takes time $O((k-1)^2 n^2)$. As $\sum_{i=1}^{k-1} i^2 = \frac{(k-1)k(2k-1)}{6}$, the overall time complexity is $O(k^3 n^2)$. Noticing that the Extended Euclidean Algorithm only needs a single reduction to get back to the faster case of two n-bit numbers and that the computation of t_1 can be carried out modulo $\frac{m_2}{g}$ decreases the computation time to $O((kn)^2)$ which is quadratic in the input size.

Remark 8.5. *Note that the bit complexity is based on the total input size. As there are k moduli, each being smaller than 2^n, the total input length is $O(kn)$. The length of the remainders can be discarded, as it would only multiply the given order by a constant factor.*

Hence, we deduce that the Existential, the Bounded, and the traditional Chinese Remainder Problem can be solved in polynomial time.

8.3 Chinese Remainder Theorem

In Section 8.2.2, the existence of solutions for a system of two congruences was based on the condition that the difference of the given remainders is divisible by the greatest common divisor of the considered moduli. Applying this condition to the multi-congruence case in Section 8.2.3 shows that if $r_i \not\equiv r_j \mod \gcd(m_i, m_j)$ for some $i, j \in \{1, \dots, k\}$, then the given Chinese Remainder Problem cannot have a solution. Conversely, if $r_i \equiv r_j \mod \gcd(m_i, m_j)$ for all $i, j \in \{1, \dots, k\}$, then our method shows how to construct a suitable solution.

Theorem 8.6 (Chinese Remainder Theorem [Coh93])**.** *Let $m_1, \dots, m_k \geq 2$ be integers, let $r_1, \dots, r_k \in \mathbb{Z}$ and let $M = \operatorname{lcm}(m_1, \dots, m_k)$. If $r_i \equiv r_j \mod \gcd(m_i, m_j)$ for all $i, j \in \{1, \dots, k\}$, then there is a unique integer $\chi \in \{0, 1, \dots, M-1\}$ such that*

$$\begin{cases} \chi & \equiv & r_1 & \mod m_1, \\ & \vdots & & \vdots \\ \chi & \equiv & r_k & \mod m_k. \end{cases}$$

If we assume the considered moduli to be pairwise coprime, the greatest common divisor condition is trivially satisfied, and so the system of congruences always has a solution. Furthermore, in this case

$$M = \operatorname{lcm}(m_1, \dots, m_k) = \prod_{i=1}^{k} m_i. \tag{8.8}$$

For the remainder of this chapter, we consider pairwise coprime moduli.

8.4 Particular solving methods

Using state-of-the-art algorithms, one can show that Chinese Remaindering with pairwise coprime moduli is quasilinear with time complexity $O(k \log^2(kn))$ [VDH16]. Hereinafter, we study some solving methods for pairwise coprime moduli. None of these methods achieves the acclaimed complexity, but they illustrate some other properties of Chinese Remaindering.

8.4.1 Systematic search

As the primitive solution to a Chinese Remainder Problem

$$\mathsf{CRP}\left((r_1, m_1), \ldots, (r_k, m_k)\right) \tag{8.9}$$

is upper bounded by M, it can be found through a systematic search. Indeed, one can start at 0 and count upwards until an integer satisfying all congruences is found. Such a procedure requires in the worst case $M = \prod_{i=1}^{k} m_i$ trials and does so not consist in an efficient solution.

8.4.2 Textbook CRT

The textbook CRT gives a remarkable closed form formula for the desired primitive solution. Indeed, if $M = \prod_{i=1}^{k} m_i$, $M_i = \frac{M}{m_i}$ and $\widetilde{M_i} = [\![M_i^{-1} \mod m_i]\!]$ for all $i \in \{1, \ldots, k\}$, then the primitive solution is given by

$$\chi = \left[\!\!\left[\sum_{i=1}^{k} r_i \widetilde{M_i} M_i \mod M \right]\!\!\right] . \tag{8.10}$$

The computation of the individual terms, as well as the final sum may involve large integers. Therefore, in practice, a recursive computation method is used.

Algorithm 8.1: Textbook CRT [Coh93]

```
Input: Given $\mathsf{CRP}((m_1, r_1), \ldots, (m_k, r_k))$
Output: The algorithm computes the primitive solution $\chi$.
$m'_1 \leftarrow m_1$;
$r'_1 \leftarrow r_1$;
for $i = 2$ until $k$ do
    $(1, t'_{i-1}, t_i) \leftarrow$Extended_Euclidean_Algorithm($m'_{i-1}$,$m_i$)
    ▷ where $(1, t'_{i-1}, t_i) \in \mathbb{Z}^3$ s.t. $1 = t'_{i-1} m'_{i-1} + t_i m_i$
    $m'_i \leftarrow m'_{i-1} m_i$;
    $r'_i \leftarrow [\![t'_{i-1} m'_{i-1} r_i + t_i m_i r'_{i-1} \mod m'_i]\!]$;
return $\chi \leftarrow r'_k$
```

8.4.3 Garner's algorithm

The runtime of the Textbook CRT, can be improved by carrying out only a single large computation. The resulting solving method is known as *Garner's algorithm* [Gar59].

Algorithm 8.2: Garner's algorithm [Coh93]

```
Input: $\mathsf{CRP}((m_1, r_1), \ldots, (m_k, r_k))$
Output: The algorithm computes the primitive solution $\chi$
$C_1 \leftarrow 1$
for $i = 2$ until $k$ do
    $C_i \leftarrow [\![(m_1 \cdots m_{i-1})^{-1} \mod m_i]\!]$
$y_1 \leftarrow [\![r_1 \mod m_1]\!]$
for $i = 2$ until $n$ do
    $y_i \leftarrow$
      $[\![r_i - (y_1 + m_1(y_2 + m_2(y_3 + (\cdots + m_{i-2}y_{i-1})\ldots)))C_i \mod m_i]\!]$
return $\chi \leftarrow (y_1 + m_1(y_2 + m_2(y_3 + (\cdots + m_{n-1}y_k)\ldots)))$
```

8.5 The power of Chinese Remaindering

The most important consequence of the Chinese Remainder Theorem is its guarantee for the existence of an efficiently computable inverse of a particular ring isomorphism.

Theorem 8.7. *Let $m_1, \ldots, m_k$ be pairwise coprime integers and let $M = \prod_{i=1}^{k} m_i$, then the map*

$$[\![x \mod M]\!] \mapsto ([\![x \mod m_1]\!], \ldots, [\![x \mod m_k]\!])$$

defines a ring isomorphism

$$\mathbb{Z}/M\mathbb{Z} \simeq \mathbb{Z}/m_1\mathbb{Z} \times \cdots \times \mathbb{Z}/m_k\mathbb{Z}.$$

Since the initial map and its inverse are efficiently computable, we can freely switch between $\mathbb{Z}/M\mathbb{Z}$ and $\mathbb{Z}/m_1\mathbb{Z} \times \cdots \times \mathbb{Z}/m_k\mathbb{Z}$. One can considerably speed up the computation runtime in $\mathbb{Z}/M\mathbb{Z}$ by first performing operations in $\mathbb{Z}/m_i\mathbb{Z}$ for each $i \in \{1, \ldots, k\}$ individually and subsequently recovering the final result in $\mathbb{Z}/M\mathbb{Z}$ using the given isomorphism. For example, this optimization strategy is used in the signing step of the famous *RSA algorithm* [RSA78] which in turn is a sub-procedure of the well known *HTTPS certification protocol*. A wide range of applications of the Chinese Remainder Theorem can be found in [DPS96].

8.6 Chinese Remaindering and Polynomial Interpolation

Polynomial Interpolation can be seen as a generalization of Chinese Remaindering where the ring of integers is replaced by a polynomial ring. Indeed, both problems recover the inverse of a particular ring isomorphism defined on a pairwise spanning system of ideals [Lip71]. To further emphasise their similarity, we recall their definitions:

- Let $m_1, \ldots, m_k$ be pairwise coprime integers, and let $r_i \in \{0, \ldots, m_i - 1\}$ for all $i \in \{1, \ldots, k\}$. The Chinese Remainder Problem asks to find an integer $x \in \mathcal{S}_M$ such that $x \equiv r_i \bmod m_i$ for all $i \in \{1, \ldots, k\}$.
- Let $x_1, \ldots, x_k$ be pairwise distinct real numbers, and let $y_i \in \mathbb{R}$ for all $i \in \{1, \ldots, k\}$. The Polynomial Interpolation Problem asks to find a polynomial $P \in \mathbb{R}[X]$ such that $P(x_i) = y_i$ for all $i \in \{1, \ldots, k\}$.

It is not surprising that solving methods for one problem are also related to the other problem. We base the upcoming development on [Sch87, Mac82].

8.6.1 Lagrange

Lagrange Interpolation constructs n polynomials such that they vanish at every point x_i except at one where the value 1 is fixed. A sum of these polynomials leads to the desired solution. To be precise, let $x_1, \ldots, x_k$ be pairwise distinct real numbers, and let $y_i \in \mathbb{R}$ for all $i \in \{1, \ldots, k\}$. Set

$$L_i(X) = \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{X - x_j}{x_i - x_j}, \tag{8.11}$$

then

$$P(X) = \sum_{i=1}^{k} y_i L_i(X) \tag{8.12}$$

is such that $P(x_i) = y_i$ for all $i \in \{1, \ldots, k\}$. The crucial part of the construction is to note that $L_i(x_j) = \delta_{i,j}$ where $\delta_{i,j}$ denotes the *Kronecker symbol* defined by $\delta_{i,j} = 1$ if $j = i$ and $\delta_{i,j} = 0$ otherwise. Applying the same trick to the Chinese Remainder Problem $\mathsf{CRP}\left((r_1, m_1), \ldots, (r_k, m_k)\right)$ leads to a non-primitive solution

$$x = \sum_{i=1}^{k} r_i \widetilde{M_i} M_i, \tag{8.13}$$

where the notations from Section 8.4.2 are used. Indeed, we note that $b_i := \widetilde{M_i} M_i \equiv \delta_{i,j} \mod m_j$ for all $i, j \in \{1, \dots, k\}$. Carrying out an additional modular reduction corresponds to Equation (8.10). Due to this connection, we classify the Textbook CRT in Section 8.4.2 as a Lagrangian solving method.

8.6.2 Newton

Newton interpolation builds the interpolation polynomial through a recursive procedure. More precisely, let $x_1, \dots, x_k$ be pairwise distinct real numbers and let $y_i \in \mathbb{R}$ for all $i \in \{1, \dots, k\}$. Set

$$\begin{array}{lcc} c_1 & = & y_1, \\ c_2 & = & \frac{y_2 - c_1}{x_2 - x_1}, \\ c_3 & = & \frac{y_3 - (c_1 + c_2(x_3 - x_1))}{(x_3 - x_1)(x_3 - x_2)}, \\ \vdots & \vdots & \vdots \\ c_k & = & \frac{y_k - \sum_{i=1}^{k}(c_i \prod_{j=1}^{i-1}(x_k - x_j))}{\prod_{i=1}^{k-1} m_i}, \end{array} \tag{8.14}$$

then,

$$\begin{aligned} P(X) &= \sum_{i=1}^{k} c_i \prod_{j=1}^{j-1} (X - x_j) && (8.15) \\ &= c_1 + c_2(X - x_1) + \cdots + c_k(X - x_1)(X - x_2)...(X - x_{k-1}) && (8.16) \\ &= c_1 + (X - x_1)(c_2 + (X - x_2)(c_3 + (\cdots + (X - x_{k-1})c_k)...)) && (8.17) \end{aligned}$$

is such that $P(x_i) = y_i$ for all $i \in \{1, \dots, k\}$. Replacing $(X - x_j)$ by m_j, and c_j by y_j for all $j \in \{1, \dots, k\}$, we observe that this procedure corresponds to Garner's algorithm in Section 8.4.3 leading to the primitive solution

$$\chi = y_1 + m_1(y_2 + m_2(y_3 + (\cdots + m_{n-1} y_k) \dots)). \tag{8.18}$$

Due to this connection, we classify Garner's algorithm as a Newtonian solving method. For completeness, we note that the improvement in Equation (8.17) is called an *Horner scheme.*

Chapter 9

Simultaneous Chinese Remaindering

Let us get back to the puzzle at the beginning of Chapter 8, but instead of considering a single remainder for each count, we put forth a set of potential remainders:

> *"There are certain things whose number is unknown. If we count them by threes, we have one or two leftovers; by fives, we have three or four leftovers; and by sevens, two, three, or five are left over. How many things are there?*

In this chapter, we first formalize the problem of finding all solutions to such a statement. Next, we investigate its relation to the traditional Chinese Remainder Problem and conclude some elementary properties. A simple counting argument shows that any method finding all solutions has an exponential runtime. Finally, we describe two elementary solving methods.

9.1 The Simultaneous Chinese Remainder Problem

The classical Chinese Remainder Problem asks to find all integer solutions to a given system of congruences where each congruence is defined by one modulus and one remainder. We consider a direct generalization of the Chinese Remainder Problem where not only a single remainder is given per modulus but each modulus is accompanied by a non-empty set of remainders.

Definition 9.1 (Simultaneous Chinese Remainder Problem (SimCRP))**.** Let $m_1, \dots, m_k \in \mathbb{N}$ be positive integers and, for all $i \in \{1, \dots, k\}$, define the nonempty set $\mathcal{R}_i \subseteq \{0, \dots, m_i - 1\}$ of remainders modulo m_i. Find all $x \in \mathcal{S}_M$, where $\mathcal{S}_M$ denotes a set of $M = \operatorname{lcm}(m_1, \dots, m_k)$ consecutive integers such that for some $(r_1, \dots, r_k) \in \mathcal{R}_1 \times \dots \times \mathcal{R}_k$

$$\begin{cases} x & \equiv & r_1 & \mod m_1, \\ & \vdots & & \vdots \\ x & \equiv & r_k & \mod m_k. \end{cases}$$

We refer to this problem by $\mathsf{SimCRP}((m_1, \mathcal{R}_1), \dots, (m_k, \mathcal{R}_k), \mathcal{S}_M)$ and call each such solution a *primitive solution.* The set of all primitive solutions is called *primitive solution set.* An integer $x' \notin \mathcal{S}_M$ satisfying such a system of congruences is called *non-primitive* solution.

As for the traditional Chinese Remainder Problem, we note that the given value restrictions do not have an impact on the generality of the problem. Indeed, any $r_i' \in \mathbb{Z}$ is congruent to some $r_i \in \{0, \dots, m_i - 1\}$ modulo m_i such that the choice of the remainder representatives does not matter. Furthermore, if there exists a solution $x \in \mathbb{Z}$ for some $(r_1, \dots, r_k) \in \mathcal{R}_1 \times \dots \times \mathcal{R}_k$, then $y \in \mathbb{Z}$ is a solution if and only if $y = x + k \operatorname{lcm}(m_1, \dots, m_k)$ for some $k \in \mathbb{Z}$. In particular, in this case, there is a unique $\chi \in \mathcal{S}_M$ that satisfies all the congruences.

Remark 9.2. *The set $\mathcal{S}_M$ in Definition 9.1 allows us to switch between different solution representatives, such as the canonical representatives $\{0, 1, \dots, M\}$ and the symmetric representatives $\mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$.*

9.2 Complexity of Simultaneous Chinese Remaindering

It is clear that a Simultaneous Chinese Remainder Problem instance

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), \dots, (m_k, \mathcal{R}_k), \mathcal{S}_M) \tag{9.1}$$

can be decomposed into $\prod_{i=1}^{k} |\mathcal{R}_i|$ traditional Chinese Remainder Problem instances. Indeed, each $(r_1, \dots, r_k) \in \mathcal{R}_1 \times \dots \times \mathcal{R}_k$ gives rise to one Chinese Remainder Problem

$$\mathsf{CRP}((m_1, r_1), \dots, (m_k, r_k), \mathcal{S}_M). \tag{9.2}$$

Put differently, traditional Chinese Remaindering is recovered by setting $|\mathcal{R}_i| = 1$ for all $i \in \{1, \ldots, k\}$. This implies that solutions to the Simultaneous Chinese Remainder Problem underlie the same existential condition as the solutions to the traditional Chinese Remainder Problem and that they can be found with the same techniques. The condition $\mathcal{R}_i \subseteq \{0, \ldots, m_i - 1\}$ for all $i \in \{1, \ldots, k\}$ implies that none of the resulting Chinese Remainder Problem instances have the same primitive solution. However, contrary to traditional Chinese Remaindering, the large number of potential solutions implies that the Simultaneous Chinese Remainder Problem cannot be solved efficiently. To be precise, the input size is linear in the number of remainders. Indeed, assuming that $m_i < 2^n$ for all $i \in \{1, \ldots, k\}$, the input size of the Simultaneous Chinese Remainder Problem is $O\left((k + \sum_{i=1}^{k} |\mathcal{R}_i|)n\right)$. On the contrary, the output size is exponential in the number of remainders. Indeed, there are up to $\prod_{i=1}^{k} |\mathcal{R}_i|$ solutions, and their representation in $\mathcal{S}_M$ is of bit-size $O(nk)$. Thus, the output size is $O\left(\left(\prod_{i=1}^{k} |\mathcal{R}_i|\right) nk\right)$. As $1 \leq |\mathcal{R}_i| \leq m_i$ for all $i \in \{1, \ldots, k\}$, the remainder set sizes $|\mathcal{R}_i|$ can be assumed to be exponential in n making so the output size exponential in nk. We note that even if $|\mathcal{R}_i| = 2$ for all $i \in \{1, \ldots, k\}$, the output size is $O\left(2^k nk\right)$ and so exponential in k.

Remark 9.3. *To recover a polynomial time solvable problem instance, only a constant number of remainder sets can be polynomial in the input size, and all the other remainder sets need to be singletons. We refer to Section 10.2.9 for another interesting observation.*

9.3 Recursive Garner

Based on the decomposability of a Simultaneous Chinese Remainder Problem instance into individual Chinese Remainder Problem instances, we conclude that we can apply any solving algorithm for Chinese Remaindering to its generalization. For example, if $m_1, \ldots, m_k$ are pairwise coprime and $S_M = \{0, \ldots, \operatorname{lcm}(m_1, \ldots, m_k) - 1\}$, then we can devise a recursive version of Garner's algorithm from Section 8.2 to find all primitive solutions.

For a given primitive solution $x' \in \mathcal{S}_{M'}$ of $\mathsf{CRP}((m_1, r_1), \ldots, (m_\ell, r_\ell), \mathcal{S}_{M'})$ where $M' = \operatorname{lcm}(m_1, \ldots, m_\ell)$ for some $1 \leq \ell < k$, we can reduce

$$\mathsf{CRP}((m_1, r_1), \ldots, (m_k, r_k), \mathcal{S}_M) \tag{9.3}$$

to

$$\mathsf{CRP}((M', x'), (m_{\ell+1}, r_{\ell+1}), \ldots, (m_k, r_k), \mathcal{S}_M). \tag{9.4}$$

Thus, a space-time trade-off allows us to decrease the number of intermediate computations.

Algorithm 9.1: Recursive Garner Algorithm

```
   Input: $\mathsf{SimCRP}((m_1, \{r_{1,1}, \dots, r_{1,t_1}\}), \dots, (m_k, \{r_{k,1}, \dots, r_{k,t_k}\}), \mathcal{S}_M)$.
   Output: The algorithm computes the primitive solution set $\mathcal{X}$.
1  $j \leftarrow 2$;
2  $C_1 \leftarrow 1$;
3  for $i = 2$ until $k$ do
4      $C_i \leftarrow \llbracket (m_1 \cdots m_{i-1})^{-1} \mod m_i \rrbracket$
5  $y_1 \leftarrow \llbracket r_{1,1} \mod m_1 \rrbracket$
6  for $i = 2$ until $k$ do
7      $y_i \leftarrow$
         $\llbracket r_{i,1} - (y_1 + m_1(y_2 + m_2(y_3 + \cdots + m_{i-2}y_{i-1}) \dots))C_i \mod m_i \rrbracket$
8  $\mathcal{X} \leftarrow \{y_1 + m_1(y_2 + m_2(y_3 + \cdots + m_{k-1}y_k) \dots)\}$
9  $(a_1, \dots, a_{k-1}, a_k) \leftarrow (1, \dots, 1, 1)$;
10 while $a_i < t_i$ for some $i = 1, \dots, k$ do
11     $a_k \leftarrow a_k + 1$;
12     $\ell \leftarrow k$
13     while $a_\ell > t_\ell$ do
14         $a_\ell \leftarrow 1$;
15         $\ell \leftarrow \ell - 1$;
16         $a_\ell \leftarrow a_\ell + 1$
17     for $i = \ell$ until $k$ do
18         $y_i \leftarrow$
             $\llbracket r_{i,a_i} - (y_1 + m_1(y_2 + m_2(y_3 + \cdots + m_{i-2}y_{i-1}) \dots))C_i \mod m_i \rrbracket$
19     $\mathcal{X} \leftarrow \mathcal{X} \cup \{y_1 + m_1(y_2 + m_2(y_3 + \cdots + m_{k-1}y_k) \dots)\}$
20 return $\mathcal{X}$
```

We observe that the precomputations (lines 1-9) only need to be carried out once and that the main loop (lines 10-19) only changes a fraction of the coefficients y_i in each iteration. The run-time can be speed-up by ordering the moduli in decreasing order and in increasing order of the remainder set sizes. Indeed, y_i is computed $\prod_{j=1}^{i} t_j$ times. Nonetheless, we cannot escape the exponential complexity described in Section 9.2 that is caused by the number of solutions.

Chapter 10

Two Simultaneous Chinese Remainder Decision Problems

This chapter is devoted to two particular Simultaneous Chinese Remainder decision problems: the *Existential* and the *Bounded* Simultaneous Chinese Remainder Problem. The Existential Simultaneous Chinese Remainder Problem asks to decide whether a given Simultaneous Chinese Remainder Problem has a solution or not. Through a direct reduction from 3-SAT, we show that this problem is $\mathsf{NP}-\mathsf{complete}$. Next, we leave the general framework and consider pairwise coprime moduli such that the existence of solutions is guaranteed. The Bounded Simultaneous Chinese Remainder Problem asks to decide whether there is a solution smaller than a given bound. Through another reduction from 3-SAT, we show that also this problem is $\mathsf{NP}-\mathsf{complete}$. Our development makes use of the complexity notions from Chapter 3 and the background work from Chapter 4.

10.1 Existential Simultaneous Chinese Remainder Problem

First, we try to decide whether solutions of a given Simultaneous Chinese Remainder Problem exist.

Definition 10.1. The *Existential* Simultaneous Chinese Remainder Problem asks to determine whether a given Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k), \mathcal{S}_M)$ has a solution or not.

We note that the moduli of the considered Simultaneous Remainder Problem may not be coprime such that the existence of solutions is non-trivial. Albeit the problem merely deviates from its classical equivalent in the number of potential remainders, the underlying decision problem experiences a significant complexity change.

Theorem 10.2. *The Existential Simultaneous Chinese Remainder Problem is* $\mathsf{NP}-\mathsf{complete}$*.*

10.1.1 Proof intuition

To show that a problem is $\mathsf{NP}-\mathsf{complete}$, we need to prove two independent properties. First, the problem needs to be in NP. This means that if a problem instance is answered affirmatively, then there needs to be a witness that can be verified in polynomial time. Thus, a polynomially verifiable witness needs to be outlined. Second, any other problem in NP needs to be reducible to the considered problem in polynomial time. It suffices to show that a particular $\mathsf{NP}-\mathsf{complete}$ problem can be reduced to the considered problem in polynomial time as the general claim follows from transitivity.

10.1.2 Proof structure

We prove Theorem 10.2 through a polynomial reduction from the 3-SAT problem to the Existential Simultaneous Chinese Remainder Problem. Our proof is subdivided into 3 parts:

A. *Membership in NP*: Proves that the Existential Simultaneous Chinese Remainder Problem is in NP.

B. *Problem construction*: Focuses on the polynomial-time construction of a SimCRP instance for a given 3-SAT instance.

C. *Solution matching*: Outlines the desired relation between solutions of the initial 3-SAT instance and the constructed SimCRP instance.

10.1.3 A. Membership in NP

Let

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k), \mathcal{S}_M) \tag{10.1}$$

be a Simultaneous Chinese Remainder Problem instance with a primitive solution x. Then, we claim that x can be used as a witness. Indeed, x has polynomial size in the Simultaneous Chinese Remainder Problem input.

Furthermore, for each $i \in \{1, \ldots, k\}$ computing $r_i := [\![x \mod m_i]\!]$ and verifying whether $r_i \in \mathcal{R}_i$ reveals in polynomial time that x is indeed a solution of the considered congruence system.

10.1.4 B. Problem construction

Let $\varphi(v_1, \ldots, v_t)$ be a Boolean expression in conjunctive normal form consisting of T clauses with 3 literals per clause. We polynomially reduce the satisfiability of φ to a specific Simultaneous Chinese Remainder Problem. We note that the input size of the given 3-SAT instance is $O(T)$ as $t \leq 3T$. Thus, for a polynomial reduction, the time and space of every upcoming computation need to be polynomial in t or T.

We start by precomputing the first $2t$ prime numbers $p_1, \ldots, p_{2t}$. By the Prime Number Theorem, we know that $p_{2t} < 2t(\log(2t) + \log(\log(2t)))$ for all $t \geq 3$ [Dus99]. Applying Eratosthenes sieve to the set

$$\mathbb{Z} \cap [2; 2t(\log(2t) + \log(\log(2t)))] \tag{10.2}$$

reveals at least $2t$ prime numbers. By [Sor98], Erathostenes sieve runs on this interval in time

$$O(t \log(t) \log(\log(t \log(t)))) \tag{10.3}$$

and space $O(t \log(t))$. Thereby, this first computation is polynomial in the 3-SAT input size.

Using these prime numbers, we design a particular Simultaneous Chinese Remainder Problem. Intuitively, we let each positive literal v_i of the given Boolean expression φ correspond to the remainder set $\mathcal{R}_i := \{0, 1\}$ modulo p_i and each negative literal $\neg v_i$ to the remainder set $\mathcal{R}_{t+i} := \{0, 1\}$ modulo p_{t+i}. However, instead of using these remainder sets individually, we consider specific polynomial size combinations. These combinations either mimic clauses of φ or simulate the logical laws of non-contradiction and the excluded middle.

First, we construct congruences corresponding to clauses of φ. We set $\ell_i := v_i$ and $\ell_{t+i} := \neg v_i$ for all $i \in \{1, \ldots, t\}$. For each clause $C_j = \ell_{j_1} \vee \ell_{j_2} \vee \ell_{j_3}$ where $j \in \{1, \ldots, T\}$ and $j_1, j_2, j_3 \in \{1, \ldots, 2t\}$, we let $m_j := p_{j_1} p_{j_2} p_{j_3}$. This multiplication is efficiently computable as the considered primes are bounded in the 3-SAT input size. Additionally, we set

$$\mathcal{R}_j := \{r_{001}, r_{010}, r_{100}, r_{110}, r_{101}, r_{011}, r_{111}\} \tag{10.4}$$

where

$$\begin{cases} r_{abc} & \equiv & a & \mod p_{j_1}, \\ r_{abc} & \equiv & b & \mod p_{j_2}, \\ r_{abc} & \equiv & c & \mod p_{j_3}. \end{cases} \tag{10.5}$$

Each such remainder set can be computed in polynomial time as is demonstrated in Section 8.2. Furthermore, a polynomial number of such remainder sets needs to be computed, namely T. Thus, we obtain in polynomial time for each clause C_j a pair $(m_j, \mathcal{R}_j)$ consisting of a modulus and its corresponding remainder set.

Remark 10.3. *A solution to the upcoming Simultaneous Chinese Remainder Problem directly yields the solution to the considered 3-SAT problem.*

Second, we construct congruences corresponding to the logical laws. For each $i \in \{1, \ldots, t\}$, we let $m_{T+i} := p_i p_{t+i}$ and $\mathcal{R}_{T+i} := \{r_{01}, r_{10}\}$ where

$$\begin{cases} r_{ab} & \equiv & a & \mod p_i, \\ r_{ab} & \equiv & b & \mod p_{t+i}. \end{cases} \tag{10.6}$$

A similar argument than above shows that each modulus and its corresponding remainder set can be computed in polynomial time. As the number of such pairs is polynomial in the input size of the 3-SAT problem, namely t, the cumulative construction is carried out in polynomial time as well.

Remark 10.4. *These congruences make sure that a solution r to the upcoming Simultaneous Chinese Remainder Problem does not violate the law of non-contradiction or the excluded middle. These laws assert that either a Boolean variable or its negation is true, but not both of them.*

Ultimately, theses polynomial time constructions lead to the following Simultaneous Chinese Remainder Problem:

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_T, \mathcal{R}_T), (m_{T+1}, \mathcal{R}_{T+1}), \ldots, (m_{T+t}, \mathcal{R}_{T+t}), \mathcal{S}_M). \tag{10.7}$$

with $\mathcal{S}_M := \{0, \ldots, \operatorname{lcm}(m_1, \ldots, m_{T+t}) - 1\} = \{0, \ldots, (\prod_{i=1}^{2t} p_i) - 1\}$.

Remark 10.5. *It doesn't matter which set of solution representatives $\mathcal{S}_M$ is chosen as the existence of a solution in one set also proves the existence of a solution in every other set of representatives. We chose the traditional representative set for simplicity only.*

10.1.5 C. Solution matching

To conclude that the above construction consists in a reduction, it suffices to show that a solution of the resulting Existential Simultaneous Chinese Remainder Problem instance yields also a solution for the initial 3-SAT problem instance. We prove that the Simultaneous Chinese Remainder Problem instance in Equation (10.7) has a solution if and only if the Boolean formula $\varphi(v_1, \ldots, v_t)$ is satisfiable. We note that this relation does not need to be efficiently computable.

If φ is satisfiable, then there is a Boolean evaluation $(v'_1, \ldots, v'_t) \in \{0,1\}^t$ such that $\varphi(v'_1, \ldots, v'_t) = 1$. We construct the following traditional Chinese Remainder Problem

$$\mathsf{CRP}((p_1, v'_1), \ldots, (p_t, v'_t), (p_{t+1}, 1 - v'_1), \ldots, (p_{2t}, 1 - v'_t), \mathcal{S}_M). \tag{10.8}$$

As the moduli are pairwise coprime, the Chinese Remainder Theorem guarantees the existence of a primitive solution $x \in \mathcal{S}_M$. We claim that x is also a primitive solution of Equation (10.7). Indeed, if x satisfies all the congruences encoded in Equation (10.7), then, as $x \in \mathcal{S}_M$, it is a primitive solution. We note that by Equation (10.8), x trivially satisfies the congruences corresponding to the logic laws in Equation (10.6). Furthermore, by replacing a, b, c in Equation (10.5) by the claimed values, we deduce that all the systems of congruences have a common non-contradictory solution.

Reciprocally, if the Simultaneous Chinese Remainder Problem instance in Equation (10.7) has a solution $x \in \mathcal{S}_M$, then, the tuple defined by $([\![x \mod p_1]\!], \ldots, [\![x \mod p_t]\!])$ is a Boolean evaluation satisfying φ. By construction, for each $j \in \{1, \ldots, k\}$, $x \equiv r_{abc} \mod m_j$ for some remainder $r_{abc} \in \mathcal{R}_j = \{r_{001}, r_{010}, r_{100}, r_{110}, r_{101}, r_{011}, r_{111}\}$. Reducing this term further with Equation (10.5) implies that

$$\begin{cases} x & \equiv & a & \mod p_{j_1}, \\ x & \equiv & b & \mod p_{j_2}, \\ x & \equiv & c & \mod p_{j_3}, \end{cases} \tag{10.9}$$

for $j_1, j_2, j_3 \in \{1, \ldots, 2t\}$ and some $a, b, c \in \{0, 1\}$ not all 0. By construction, it is clear that $[\![x \mod p_i]\!] \in \{0,1\}$ for all $i \in \{1, \ldots, t\}$ revealing so (binary) Boolean values. As at least one of a, b, c is non-zero, the corresponding clause $C_j = [\![x \mod p_{j_1}]\!] \vee [\![x \mod p_{j_2}]\!] \vee [\![x \mod p_{j_3}]\!]$ is satisfied. As this conclusion holds for each clause of φ, φ is satisfied. It remains to show that there is no contradiction. This is taken care of by the uniqueness of Chinese Remaindering and the fact that x satisfies Equation (10.6) such that a variable and its negative cannot have the same value.

□

10.2 Bounded Simultaneous Chinese Remainder Problem

As Theorem 10.2 shows that deciding whether a given Simultaneous Chinese Remainder Problem

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k), \mathcal{S}_M) \tag{10.10}$$

has a solution or not is $\mathsf{NP} - \mathsf{complete}$, finding any solution, if it exists is at least as hard. To simplify the problem, we may restrict to pairwise coprime moduli $m_1, \ldots, m_k$. In this case, the traditional Chinese Remainder Theorem proves the existence of $\prod_{i=1}^{k} |\mathcal{R}_i|$ solutions in $\mathcal{S}_M$. The Existential Chinese Remainder Problem becoming trivial, we may ask about the size of solutions.

Remark 10.6. *The notion of size in this setting either means the smallest element in the chosen set of representatives $\mathcal{S}_M \subseteq \mathbb{Z}$ or the smallest element in $\mathcal{S}_M$ in absolute value. Whereas the former notion is absolute, the latter notion strongly depends on the choice of the representative set. Despite the seeming ambiguity, Section 10.2.10 shows that it makes no difference.*

Hereinafter, we consider the particular representative set $\mathcal{S}_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$ and the size in absolute value.

Definition 10.7. The *Bounded* Simultaneous Chinese Remainder Problem asks to determine whether a given Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k), \mathcal{S}_M)$, with pairwise coprime moduli $m_1, \ldots, m_k$ and representative set $\mathcal{S}_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$, has a solution $|x| < B$ for some given bound $B \in \mathcal{S}_M$ such that $0 < B < \frac{M}{2}$ where $M = \prod_{i=1}^{k} m_i$.

The following result shows that the Bounded Simultaneous Chinese Remainder Problem is hard in general.

Theorem 10.8. *The Bounded Simultaneous Chinese Remainder Problem is* $\mathsf{NP}{-}\mathsf{complete}$.

10.2.1 Proof intuition

The proof intuition is the same as in Section 10.1.1.

10.2.2 Comment

[MA76] states that deciding whether for given positive integers a, b, c there is a positive integer $x < c$ such that $x^2 \equiv a \mod b$ is $\mathsf{NP-complete}$ even if the prime decomposition $b = \prod_{i=1}^{k} p_i^{\alpha_i}$ is known and solutions to $x^2 \equiv a \mod p_i^{\alpha_i}$ for all $i \in \{1, \dots, k\}$ are given. The Bounded Simultaneous Chinese Remainder Problem naturally extends this problem to quadratic non-residues. Therefore, we could reduce this problem on quadratic residues to the Bounded Simultaneous Chinese Remainder Problem and conclude its $\mathsf{NP-completeness}$. Yet, hereinafter, we develop a direct reduction from 3-SAT. We follow closely the development in [MA76].

10.2.3 Proof structure

Our proof outlines a polynomial reduction from the 3-SAT problem to the Bounded Simultaneous Chinese Remainder Problem. It is subdivided into 3 parts:

A. *Membership in NP*: Proves that the Bounded Simultaneous Chinese Remainder Problem is in NP.

B. *Problem construction*: Focuses on the polynomial-time construction of a SimCRP instance for a given 3-SAT instance.

C. *Solution matching*: Outlines the desired relation between solutions of the initial 3-SAT instance and the constructed SimCRP instance.

Albeit the proof structure is the same as for Theorem 10.2, the problem construction is highly different and requires some intermediate observations.

10.2.4 A. Membership in NP

Let

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k), \mathcal{S}_M) \tag{10.11}$$

be a Simultaneous Chinese Remainder Problem instance with a primitive solution x such that $|x| < B$. Then, we claim that x can be used as a witness. Indeed, x has polynomial size in the Simultaneous Chinese Remainder input. Furthermore, for each index $i \in \{1, \dots, k\}$ computing $r_i = [\![x \mod m_i]\!]$ and verifying whether $r_i \in \mathcal{R}_i$ reveals in polynomial time that x is indeed a solution of the considered congruence system. An auxiliary linear comparison with B shows that $|x| < B$.

10.2.5 B. Problem construction

Let $\varphi(v_1, \ldots, v_t)$ be a Boolean expression in conjunctive normal form consisting of T_φ clauses with 3 literals per clause. Without loss of generality, we can assume that φ does not contain clauses that contain simultaneously a variable v_i and is negation $\neg v_i$ for some $i \in \{1, \ldots, t\}$. Otherwise, we can reduce φ in linear time to such an expression by simply ignoring those clauses as they are anyways satisfied. Similarly, we can assume that φ does not contain duplicate clauses as they are either all satisfied or all not satisfied. We note that in such a reduced Boolean expression, there are less than $T = (2t)^3$ distinct ordered clauses. We note that we distinguish clauses with the same literals but in a different order. For example $v_1 \vee v_2 \vee v_3$ and $v_2 \vee v_1 \vee v_3$ are distinguished (albeit only one of them is contained in ϕ by a previous assumption). Let $(C_1, \ldots, C_T)$ be an enumeration of all ordered clauses that can be formed in this way. We say that a literal ℓ belongs to a clause C and denote it by $\ell \in C$ if ℓ is the first, second, or third literal of the ordered clause. Similarly, we say that an ordered clause C belongs to the Boolean expression φ and denote it by $C \in \varphi$ if C occurs among the clauses of φ. Our first objective is to construct a Simultaneous Chinese Remainder Problem corresponding to the Boolean expression φ.

We note that by assumption each ordered clause C_j for $j \in \{1, \ldots, T\}$ appears at most once among the clauses of φ. To encode which clauses belong to φ, we set for each $j \in \{1, \ldots, T\}$ the value

$$\epsilon_j := \begin{cases} 1 & \text{if } C_j \in \varphi \\ 0 & \text{if } C_j \notin \varphi \end{cases} \tag{10.12}$$

The cumulative information on those ϵ_j can be stored as an integer in base 8 notation

$$\tau_\varphi := -\sum_{j=1}^{T} \epsilon_j 8^j. \tag{10.13}$$

This computation is polynomial in the 3-SAT input size. Indeed, the generation of the ordered list of clauses is cubic in t and running once through φ to assign the ϵ_j values is linear in the input size of φ. Finally, $-\tau_\varphi \leq \frac{8^{T+1}-1}{8-1} - 1 < 8^{T+1}$ making sure that the bit length of the result and its intermediate values are polynomially bounded in the 3-SAT input size.

Next, we wish to encode which literals belong to the enumerated clauses. To do so, we use the same base 8 representation approach. More precisely, for each $i \in \{1, \dots, t\}$, we set

$$f_i^+ = \sum_{j=1}^{T} \mathbb{1}_{C_j}(v_i)\, 8^j \quad \text{and} \quad f_i^- = \sum_{j=1}^{T} \mathbb{1}_{C_j}(\neg v_i)\, 8^j \tag{10.14}$$

where $\mathbb{1}_C$ denotes the indicator function defined by $\mathbb{1}_C(v) = 1$ if $v \in C$ and $\mathbb{1}_C(v) = 0$ if $v \notin C$. Again, these computations can be carried out in polynomial time.

Subsequently, we can start to devise the Simultaneous Chinese Remainder Problem. We set

$$n = 2T + t \tag{10.15}$$

and compute the first $n+1$ primes $p_0, \dots, p_n$ larger than 12. We note that n is a polynomial value in T. As described in Section 10.1.4 the computation of such primes is polynomial in n and the largest prime has size $O(n \log(n))$.

The pairwise coprime moduli of the Simultaneous Chinese Remainder Problem are obtained by setting $m_{-1} := 8^{T+1}$ and for all $j \in \{0, \dots, n\}$ setting

$$m_j := p_j^{n+1}. \tag{10.16}$$

It is important to note that these moduli are still polynomial in the 3-SAT input size. Indeed, the bit size of the largest modulus is $O(n^2 \log(n))$. Furthermore, for later use, we set

$$M := \prod_{i=0}^{n} m_i \quad \text{and} \quad M_j := \frac{M}{m_j}. \tag{10.17}$$

M is at most $n+1$ times larger than the largest modulus and thus of bit size $O(n^3 \log(n))$.

Remark 10.9. *A polynomial number of multiplications of polynomial bit-sized integers leads to a polynomial bit-sized integer.*

For the remainder sets and the bound of the Bounded Simultaneous Chinese Remainder Problem, we define first the following parameters:

- For $j = 0$, let
$$\lambda_0 := [\![M_0^{-1} \mod 8^{T+1}]\!]. \tag{10.18}$$
- For $j \in \{1, \dots, T\}$ let
$$\lambda_{2j-1} := \left[\!\!\left[-\frac{1}{2} 8^j M_{2j-1}^{-1} \mod 8^{T+1} \right]\!\!\right] \tag{10.19}$$
and
$$\lambda_{2j} := \left[\!\!\left[-8^j M_{2j-1}^{-1} \mod 8^{T+1} \right]\!\!\right]. \tag{10.20}$$

- For $j \in \{2T+1, \dots, 2T+t\}$ let

$$\lambda_j := \left[\!\!\left[\frac{1}{2} \left(f_{j-2T}^{+} - f_{j-2T}^{-} \right) \mod 8^{T+1} \right]\!\!\right]. \tag{10.21}$$

We note that the inverses above are taken with respect to the modulus 8^{T+1} and that the parameter in Equation (10.21) is well defined as f_{j-2T}^{+} and f_{j-2T}^{-} are divisible by 8 for all $j \in \{2T+1, \dots, 2T+t\}$. The computation of these parameters requires the use of the Extended Euclidean Algorithm on integers of polynomial size in the 3-SAT input. Thus, each computation is polynomial in the 3-SAT input size. We define the bound

$$B := \sum_{j=0}^{n} \lambda_j M_j, \tag{10.22}$$

the remainder set

$$\mathcal{R}_{-1} := \left\{ \left[\!\!\left[-\tau_{\varphi} - B - \sum_{i=1}^{t} f_i^{-} \mod 8^{T+1} \right]\!\!\right] \right\} \tag{10.23}$$

and for each $j \in \{0, \dots, n\}$ the remainder set

$$\mathcal{R}_j := \{ [\![-\lambda_j M_j \mod m_j]\!], [\![\lambda_j M_j \mod m_j]\!] \} \tag{10.24}$$

all of which are computable in polynomial time from the given parameters.

Ultimately, theses polynomial time constructions lead to the following Simultaneous Chinese Remainder Problem:

$$\mathsf{SimCRP}((m_{-}1, \mathcal{R}_{-1}), \dots, (m_n, \mathcal{R}_n), \mathcal{S}_{M'}). \tag{10.25}$$

with $\mathcal{S}_{M'} := \mathbb{Z} \cap \left(-\frac{8^{T+1}M}{2}, \frac{8^{T+1}M}{2} \right]$ where $M = \prod_{i=0}^{n} m_i$, and the solution bound $B = \sum_{j=0}^{n} \lambda_j M_j$.

10.2.6 Bound analysis

We note that the solution bound B in Equation (10.22) satisfies the required condition $B < \frac{8^{T+1}M}{2}$. Indeed,

$$B < \sum_{j=0}^{n} 8^{T+1} M_j = \sum_{j=0}^{n} 8^{T+1} \frac{M}{p_j^{n+1}}. \tag{10.26}$$

By construction, we have $p_j > 12$ for all $j \in \{0, \dots, n\}$. Furthermore,

$$12 > \left(4(n+1)8^{T+1}\right)^{\frac{1}{n+1}} . \tag{10.27}$$

Indeed, this equation is equivalent to

$$\frac{12^{n+1}}{8^{T+1}} > 4(n+1). \tag{10.28}$$

Using $n = 2T + t$, we observe that the left hand side is equal to

$$\frac{12^{n+1}}{8^{T+1}} = 12^{t-1}18^{T+1}, \tag{10.29}$$

such that Equation (10.28) is equivalent to

$$12^{t-1}\, 18^{T+1} > 4(2T + T + 1). \tag{10.30}$$

An elementary function analysis proves Equation (10.30) which implies Equation (10.27). Thus, Equation (10.26) becomes

$$B < \sum_{j=0}^{n} 8^{T+1} \frac{M}{4(n+1)8^{T+1}} = \frac{M}{4} < \frac{8^{T+1}M}{2} \tag{10.31}$$

as required by the Bounded Simultaneous Chinese Remainder Problem.

10.2.7 Some observations

First, we note that $x \in \mathcal{S}_{M'}$ is a solution of the Simultaneous Chinese Remainder Problem instance in Equation (10.25) if and only if

$$\begin{cases} x \;\equiv\; -\tau_\varphi - B - \sum_{i=1}^{t} f_i^- & \bmod 8^{T+1}, \\ x \;\equiv\; \sum_{j=0}^{n} \alpha_j \lambda_j M_j & \bmod M, \end{cases} \tag{10.32}$$

where $\alpha_j \in \{-1, 1\}$ for all $j \in \{0, \dots, n\}$. Indeed, any such x is a solution as the first congruence satisfies Equation (10.23) and the second congruence satisfies Equation (10.24). Furthermore, a trivial comparison shows that for varying α_j each such solution differs in at least one congruence and their total number is 2^{n+1} which is exactly the number of solutions expected for the problem. Thus, any solution needs to be of this form.

By the second congruence in Equation (10.32), we deduce that any solution satisfies

$$x = \sum_{j=0}^{n} \alpha_j \lambda_j M_j + kM \tag{10.33}$$

for some $k \in \mathbb{Z}$. As $B = \sum_{j=0}^{n} \lambda_j M_j$, it trivially holds that

$$-B \leq \sum_{j=0}^{n} \alpha_j \lambda_j M_j \leq B \tag{10.34}$$

and by our observation in Equation (10.31), we know that $B < \frac{M}{2}$ showing that $-B < x < B$ if and only if $k = 0$. Thus, x is a solution to the Simultaneous Chinese Remainder Problem instance in Equation (10.25) if and only if

$$\begin{cases} x & \equiv & -\tau_\varphi - B - \sum_{i=1}^{t} f_i^- & \mod 8^{T+1}, \\ x & = & \sum_{j=0}^{n} \alpha_j \lambda_j M_j, & \end{cases} \tag{10.35}$$

for some $\alpha_j \in \{-1, 1\}$ for all $j \in \{0, \ldots, n\}$.

Inserting the expression $x = \sum_{j=0}^{n} \alpha_j \lambda_j M_j$ and $B = \sum_{j=0}^{n} \lambda_j M_j$ into the first congruence of Equation (10.35) leads to

$$\tau_\varphi + \sum_{i=1}^{t} f_i^- + \sum_{j=0}^{n} (\alpha_j + 1) \lambda_j M_j \equiv 0 \mod 8^{T+1}. \tag{10.36}$$

Setting for all $j \in \{0, \ldots, n\}$ $\beta_j := 0$ if $\alpha_j = -1$ and $\beta_j := 1$ if $\alpha_j = 1$ yields

$$\tau_\varphi + \sum_{i=1}^{t} f_i^- + \sum_{j=0}^{n} 2\beta_j \lambda_j M_j \equiv 0 \mod 8^{T+1}. \tag{10.37}$$

Using the definitions of $\lambda_0, \ldots, \lambda_n$, we deduce that

$$\sum_{j=0}^{n} 2\beta_j \lambda_j M_j \tag{10.38}$$

$$\equiv 2\beta_0 \lambda_0 M_0 + \sum_{j=1}^{T} (2\beta_{2j-1} \lambda_{2j-1} M_{2j-1} + 2\beta_{2j} \lambda_{2j} M_{2j})$$

$$+ \sum_{j=2T+1}^{2T+t} 2\beta_j \lambda_j M_j \mod 8^{T+1} \tag{10.39}$$

$$\equiv 2\beta_0 + \sum_{j=1}^{T} (-\beta_{2j-1} 8^j - 2\beta_{2j} 8^j) + \sum_{j=1}^{t} \beta_{2T+j} (f_j^+ - f_j^-) \mod 8^{T+1} \tag{10.40}$$

Using the expression in Equation (10.40) and replacing $\tau_\varphi, f_i^+, f_i^-$ by their respective representations from Equation (10.13) and Equation (10.14), Equation (10.37) becomes

$$2\beta_0 + \sum_{j=1}^{T} \left(-\epsilon_j - \beta_{2j-1} - 2\beta_{2j} + \sum_{i=1}^{t} \beta_{2T+i} \mathbb{1}_{C_j}(v_i) + \sum_{i=1}^{t} (1 - \beta_{2T+i}) \mathbb{1}_{C_j}(\neg v_i) \right) 8^j$$
$$\equiv 0 \mod 8^{T+1}. \tag{10.41}$$

Setting

$$R_j := -\epsilon_j - \beta_{2j-1} - 2\beta_{2j} + \sum_{i=1}^{t} \beta_{2T+i} \mathbb{1}_{C_j}(v_i) + \sum_{i=1}^{t} (1 - \beta_{2T+i}) \mathbb{1}_{C_j}(\neg v_i) \tag{10.42}$$

for all $j \in \{1, \ldots, T\}$, Equation (10.41) becomes

$$2\beta_0 + \sum_{j=1}^{T} R_j 8^j \equiv 0 \mod 8^{T+1}. \tag{10.43}$$

We note that $\beta_0 \in \{0, 1\}$ and for each $j \in \{1, \ldots, T\}$,

$$-4 \leq R_j \leq 3. \tag{10.44}$$

Indeed, as each clause is composed of 3 literals the sum of the two sums in Equation (10.42) of R_j belongs to $\{0, 1, 2, 3\}$ explaining the upper bound. Furthermore, $\epsilon_j \in \{0, 1\}$ and $\beta_j \in \{0, 1\}$ leading to the lower bound.

We note that the congruence in Equation (10.43) holds and so x is a solution to the Simultaneous Chinese Remainder Problem instance in Equation (10.25) if and only if $\beta_0 = 0$ and

$$R_j = 0 \tag{10.45}$$

for all $j \in \{1, \ldots, T\}$. Concretely, by the bounds on R_j, we have

$$2\beta_0 + \sum_{j=1}^{T} R_j 8^j \geq 0 - 4 \sum_{j=1}^{T} 8^j = -\frac{4}{7}(8^{T+1} - 8) > -8^{T+1} \tag{10.46}$$

$$2\beta_0 + \sum_{j=1}^{T} R_j 8^j \leq 2 + 3 \sum_{j=1}^{T} 8^j = \frac{3}{7}(8^{T+1} - 8) < 8^{T+1} \tag{10.47}$$

and so the congruence in Equation (10.43) is satisfied if and only if

$$2\beta_0 + \sum_{j=1}^{T} R_j 8^j = 0 \tag{10.48}$$

over the integers. The same argument in a downwards induction loop from T to 1 shows that Equation (10.48) holds if and only if $\beta_0 = 0$ and $R_j = 0$ for all $j \in \{1, \ldots, T\}$.

10.2.8 C. Solution matching

To conclude that our construction in Section 10.2.5 consists in a reduction, it remains to show that a solution to the constructed Bounded Simultaneous Chinese Remainder Problem instance yields also a solution to the initial 3-SAT problem instance. We prove that the Simultaneous Chinese Remainder Problem instance in Equation (10.25) has a primitive solution bounded by B if and only if the Boolean formula $\varphi(v_1, \ldots, v_t)$ is satisfiable. We note that this relation does not need to be efficiently computable.

We relate solutions in the following manner:

- for all $i \in \{1, \ldots, t\}$, $v_i = \beta_{2T+i}$,
- for all $j \in \{1, \ldots, T\}$, $\beta_{2j-1} + 2\beta_{2j} + \epsilon_j$ equals the number of 1 valued literals in C_j,
- for $j = 0$, $\beta_0 = 0$.

If there exists a solution $x \in \mathcal{S}_{M'}$ of the Simultaneous Chinese Remainder Problem in Equation (10.25) such that $|x| < B$, then, by Equation (10.45), $R_j = 0$ for all $j \in \{1, \ldots, T\}$. Let $k \in \{1, \ldots, T\}$ be the index of any clause $C_k \in \varphi$. Then, by definition $\epsilon_k = 1$ and so, as $R_k = 0$, Equation (10.42) implies

$$\sum_{i=1}^{t} \beta_{2T+i} \mathbb{1}_{C_j}(v_i) + \sum_{i=1}^{t} (1 - \beta_{2T+i}) \mathbb{1}_{C_j}(\neg v_i) > 0 \tag{10.49}$$

Thus, either $\beta_{2T+i} = 1$ and $v_i \in C_j$, or $\beta_{2T+i} = 0$ and $\neg v_i \in C_j$. In both cases, by the particular choice of the value v_i, the clause C_j is satisfied. As this holds for every clause, φ is satisfied. Additionally, we observe that $\beta_{2j-1} + 2\beta_{2j} + \epsilon_j$ corresponds to the number of 1 valued literals in C_j. Indeed, otherwise, $R_j \neq 0$. Furthermore, $\beta_0 = 0$ by construction.

Reciprocally, assume that $(v'_1, \ldots, v'_t)$ is a list of values that satisfies φ. Then, a direct construction based on Equation (10.42) shows that for each $k \in \{0, \ldots, 2T + t\}$ the value of β_k is fixed such that $R_j = 0$ for all $j \in \{1, \ldots, T\}$. Indeed, for each $i \in \{0, \ldots, 2T + t\}$ the value of β_i is completely determined by the above relations and they imply that $\sum_{i=1}^{t} \beta_{2T+i} \mathbb{1}_{C_j}(v_i) + \sum_{i=1}^{t} (1 - \beta_{2T+i}) \mathbb{1}_{C_j}(\neg v_i) = \epsilon_j + \beta_{2j-1} + 2\beta_{2j}$. Inserting β_j into α_j in Equation (10.35) leads immediately to a solution to the Bounded Simultaneous Chinese Remainder Problem in Equation (10.25). □

10.2.9 The hardness of different instances

We highlight that the development above reduces any 3-SAT instance to a Bounded Simultaneous Chinese Remainder Problem instance with a particular form. Concretely, the considered Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_-1, \mathcal{R}_{-1}), \ldots, (m_n, \mathcal{R}_n), \mathcal{S}_{M'})$ in Equation (10.25) consists of remainder sets with at most two elements. Indeed, $|\mathcal{R}_{-1}| = 1$ and $|\mathcal{R}_i| = 2$ for all $i \in \{0, \ldots, n\}$.

Intuitively, the problem seems to get easier if the remainder set sizes grow: there are more solutions, and so, heuristically, the probability of finding small solutions grows. Thus, for a sufficiently large bound B, the problem should become trivial. This intuition is backed up in Chapter 12 where a rough upper bound for the minimal solution is developed.

Strangely, the problem also becomes easier if fewer remainders are used. Indeed, if the remainder sets contain only a single element, then we recover the Bounded Chinese Remainder Problem that can be solved in polynomial time. Similarly, problem instances with slightly larger remainder sets can be solved polynomially as is described in Remark 9.3.

10.2.10 A generalization

Although the Bounded Simultaneous Chinese Remainder Problem has been announced with respect to the solution representation set $\mathcal{S}_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$ and the size in absolute value, we point out that this arbitrary choice does not impact the conclusion. Indeed, we may consider the following generalization.

Definition 10.10. The *General Bounded* Simultaneous Chinese Remainder Problem asks to determine whether a given Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k), \mathcal{S}_M)$, with pairwise coprime moduli $m_1, \ldots, m_k$ has a solution $x \in \mathcal{S}_M$ such that $x < B$ for some predefined $B \in \mathcal{S}_M$.

Through an elementary shift of the solution set, we obtain the following result.

Theorem 10.11. *The General Bounded Simultaneous Chinese Remainder Problem with a fixed representative set $\mathcal{S}_M$ is polynomially equivalent to the Bounded Simultaneous Chinese Remainder Problem.*

Proof. Let

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k), \mathcal{S}_M) \tag{10.50}$$

with bound $B \in \mathcal{S}_M$ be a given General Bounded Simultaneous Chinese Remainder Problem instance. We devise a polynomial reduction to the Bounded Simultaneous Chinese Remainder Problem instance. Indeed, we compute $\beta := \frac{B+\min \mathcal{S}_M}{2}$. Next, we set $\mathcal{R}'_i := \{[\![r - \lfloor \beta \rfloor \mod m_i]\!] \mid r \in \mathcal{R}_i\}$ for all $i \in \{1, \ldots, k\}$ and $B' := B - \lceil \beta \rceil$. Finally, we construct the Bounded Simultaneous Chinese Remainder Problem instance

$$\mathsf{SimCRP}((m_1, \mathcal{R}'_1), ..., (m_k, \mathcal{R}'_k), \mathcal{S}'_M) \tag{10.51}$$

with bound B' where $\mathcal{S}'_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$ and $M = \prod_{i=1}^k m_i$. We note that all of these computations are linear in the input size. Furthermore, any solution $x' \in \mathcal{S}'_M$ to the Bounded Simultaneous Chinese Remainder Problem in Equation (10.51) yields a solution $x \in \mathcal{S}_M$ to the General Bounded Simultaneous Chinese Remainder Problem in Equation (10.50) by setting $x := x' + \lfloor \beta \rfloor$. By construction $-B' < x' < B'$ and so replacing $B' = B - \lceil \beta \rceil$ yields

$$-B + \lceil \beta \rceil + \lfloor \beta \rfloor < x' + \lfloor \beta \rfloor < B - \lceil \beta \rceil + \lfloor \beta \rfloor. \tag{10.52}$$

As $\beta = \frac{B+\min \mathcal{S}_M}{2}$, the left hand side equals $\min \mathcal{S}_M$ and the right hand side equals B if $\beta \in \mathbb{Z}$ and $B - 1$, otherwise. Thus, $x \in \mathcal{S}_M$ and $x < B$. As $[\![x' \mod m_i]\!] \in \mathcal{R}'_i$, also

$$[\![x \mod m_i]\!] = [\![x' + \lfloor \beta \rfloor \mod m_i]\!] \in \mathcal{R}_i \tag{10.53}$$

by construction. Thus, x is indeed a solution to the General Bounded Simultaneous Chinese Remainder Problem in Equation (10.50). Conversely, if the General Bounded Simultaneous Chinese Remainder Problem in Equation (10.50) has a solution other than $B - 1$, then also the Bounded Simultaneous Chinese Remainder Problem in Equation (10.51) has a solution whose existence can be shown through a similar development. However, if $\beta \notin \mathbb{Z}$, then the above reduction is falling short of the potential solution $B - 1$ which needs to be tested individually in polynomial time.

Reciprocally, let

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k), \mathcal{S}'_M) \tag{10.54}$$

with a bound $B \in \mathcal{S}'_M$ such that $0 < B < \frac{M}{2}$ where $\mathcal{S}'_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$ and $M = \prod_{i=1}^k m_i$ be a given Bounded Simultaneous Chinese Remainder Problem instance. We devise a polynomial reduction to a General Bounded Simultaneous Chinese Remainder Problem instance for a fixed set of solution representatives $\mathcal{S}_M$. $\mathcal{S}_M$ needs to be part of the problem input as the set $\mathcal{S}_M$

may have an exponential representation, whose computation could hinder a polynomial-time reduction. We compute $\beta := B + \min \mathcal{S}_M$. Next, we set

$$\mathcal{R}'_i := \{[\![r + \beta \mod m_i]\!] \mid r \in \mathcal{R}_i\} \tag{10.55}$$

for all $i \in \{1, \dots, k\}$ and $B' := B + \beta$. Finally, we construct the General Bounded Simultaneous Chinese Remainder Problem instance

$$\mathsf{SimCRP}((m_1, \mathcal{R}'_1), ..., (m_k, \mathcal{R}'_k), \mathcal{S}_M) \tag{10.56}$$

A similar development as above shows that any solution x to the General Bounded Simultaneous Chinese Remainder Problem in Equation (10.56) reveals a solution $x' = x - \beta$ to the Bounded Simultaneous Chinese Remainder Problem in Equation (10.54) and that if the Bounded Simultaneous Chinese Remainder Problem in Equation (10.54) has a solution, so does the General Bounded Simultaneous Chinese Remainder Problem in Equation (10.56). □

Clearly, the General Bounded Simultaneous Chinese Remainder Problem is in NP as a solution $x < B$ to a given Simultaneous Chinese Remainder Problem instance problem instance can be used as a polynomially verifiable witness. Thereby, we conclude the following corollary.

Corollary 10.12. *The General Bounded Simultaneous Chinese Remainder Problem is* $\mathsf{NP}-\mathsf{complete}$.

Chapter 11

Simultaneous Chinese Remainder Problem variants

In Section 9.2 we showed that the Simultaneous Chinese Remainder Problem can in general only be solved in exponential time in its input size. Theorem 10.2 yields that even deciding whether there exit solutions or not is $\mathsf{NP}-\mathsf{complete}$. Simplifying the problem by considering pairwise coprime moduli such that solutions are guaranteed to exist trivially solves the former problem but raises the question about the size of the smallest solution. Theorem 10.8 proves that if the solution set $\mathcal{S}_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$ is considered then deciding whether a solution of a given size exists is again $\mathsf{NP}-\mathsf{complete}$. Furthermore, Theorem 10.11 yields that the considered solution set $\mathcal{S}_M$ does not have an impact on the complexity of the problem.

In this chapter, we focus on finding solutions with particular properties. For simplicity, we still consider pairwise coprime moduli $m_1, \ldots, m_k$ and we restrict to the traditional representative set $\mathcal{S}_M = \{0, \ldots, M-1\}$ with $M = \prod_{i=1}^{k} m_i$. Hereinafter we omit $\mathcal{S}_M$ from our notation.

11.1 The Minimal Simultaneous Chinese Remainder Problem

We first concentrate on finding the minimal solution to a Simultaneous Chinese Remainder Problem.

Definition 11.1. The *Minimal* Simultaneous Chinese Remainder Problem asks to determine the minimal solution to a given Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$.

As Corollary 10.12 proves that the General Bounded Simultaneous Chinese Remainder Problem is $\mathsf{NP}-\mathsf{complete}$, and as the Minimal Simultaneous Chinese Remainder Problem trivially solves the General Bounded Simultaneous Chinese Remainder Problem, the Minimal Simultaneous Chinese Remainder Problem must be at least equally hard.

Theorem 11.2. *The Minimal Simultaneous Chinese Remainder Problem is* $\mathsf{NP}-\mathsf{hard}$*.*

We note that any solution to a Minimal Simultaneous Chinese Remainder Problem can be verified in polynomial time.

11.2 Other related problems

In the same mindset as above, we can construct other problems asking for solutions of a specific size.

Definition 11.3. The *Maximal* Simultaneous Chinese Remainder Problem asks to determine the maximal solution to a given Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$.

Definition 11.4. The *Interval* Simultaneous Chinese Remainder Problem asks to determine, if possible, a solution x to a given Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$ inside a given interval $I := [a, b]$ with $a, b \in \mathbb{Z} \cap [0, \prod_{i=1}^k m_i)$.

Remarkably, none of these problems is substantially harder than the Minimal Simultaneous Chinese Remainder Problem as shows the following proposition.

Proposition 11.5. *The Minimal, Maximal, and Interval Simultaneous Chinese Remainder Problems can be reduced to each other in polynomial time.*

Proof. The general approach of the reductions is still the same as in Chapter 10, but we adopt an algorithmic perspective using problem solvers as sub-routines.

1. First, we show that the Minimal and Maximal Simultaneous Chinese Remainder Problems are linearly equivalent:

   (a) Let $\mathcal{A}$ be an algorithm to solve Minimal Simultaneous Chinese Remainder Problem instances. We construct a linear time reduction to find the maximal solution of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$:

i. Change the remainder sets to

$$\overline{\mathcal{R}_i} := \{[\![-r \bmod m_i]\!] \mid r \in \mathcal{R}_i\}. \tag{11.1}$$

ii. Use algorithm $\mathcal{A}$ to find the minimal solution $\overline{\chi_{min}}$ of

$$\mathsf{SimCRP}((m_1, \overline{\mathcal{R}_1}), ..., (m_k, \overline{\mathcal{R}_k})). \tag{11.2}$$

iii. Return $\chi_{\max} = \left[\!\!\left[-\overline{\chi_{\min}} \mod \prod_{i=1}^{k} m_i \right]\!\!\right]$.

We note that the first step carries out $|\mathcal{R}_i|$ modular reductions for each $i \in \{1, \ldots, k\}$. Each such reduction is linear in m_i as a single addition is required. The last step requires a single reduction linear in the input size. For correctness, we need to show that $\chi_{\max}$ is indeed the maximal primitive solution of the initial Simultaneous Chinese Remainder Problem instance. Assume by contradiction that it is not. Then, there is another solution x such that $\chi_{\max} < x$. This implies that

$$\left[\!\!\left[-\chi_{\max} \mod \prod_{i=1}^{k} m_i \right]\!\!\right] > \left[\!\!\left[-x \mod \prod_{i=1}^{k} m_i \right]\!\!\right] \tag{11.3}$$

contradicting the minimality of $\overline{\chi_{min}}$.

(b) A similar development shows the converse implication.

2. Second, we show that the Minimal Simultaneous Chinese Remainder Problem is quasi-linearly equivalent to the Interval Simultaneous Chinese Remainder Problem:

(a) Let $\mathcal{A}$ be an algorithm to solve the Minimal Simultaneous Chinese Remainder Problem. We construct a linear time reduction to find, if it exists, a solution of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$ in $I := [a, b]$ with $a, b \in \mathbb{Z} \cap [0, \prod_{i=1}^{k} m_i[$:

i. Change the remainder sets to

$$\overline{\mathcal{R}_i} = \{[\![r - a \mod m_i]\!] \mid r \in \mathcal{R}_i\}. \tag{11.4}$$

ii. Use the algorithm $\mathcal{A}$ to find the minimal solution

$$\overline{\chi_{min}} = \mathsf{SimCRP}_{\min}((m_1, \overline{\mathcal{R}_1}), ..., (m_k, \overline{\mathcal{R}_k})). \tag{11.5}$$

iii. Compute $\chi_I = \left[\!\!\left[\overline{\chi_{\min}} + a \mod \prod_{i=1}^{k} m_i \right]\!\!\right]$.

iv. If $\chi_I < b$, return χ_I, else return null.

Once again, linear time follows from the efficiency of modular arithmetic and correctness stems from the minimality of $\chi_{\min}$.

(b) Let $\mathcal{A}$ be an efficient algorithm to solve the Interval Simultaneous Chinese Remainder Problem. We construct a quasi-linear time reduction to find the minimal solution of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$:

i. Set $a = 0$ and $b = \prod_{i=1}^{k} m_i - 1$ and let $I = [a, b]$.

ii. If $b - a < 1$, return $I \cap \mathbb{N}$, else continue.

iii. Set $c = \frac{b-a}{2}$ and set $I' = [a, c]$.

iv. Use $\mathcal{A}$ to solve $\mathsf{SimCRP}_{I'}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$ either resulting in a solution χ or null.

v. If a solution χ is obtained set $I = [a, c]$, else set $I = [c, b]$ restart from (ii).

This procedure is dominated by the number of calls to $\mathcal{A}$ which is $\log_2((\prod_{i=1}^{k} m_i) - 1)$ leading to a quasi-linear time reduction. Correctness stems directly from the construction.

□

11.3 Problems in the literature

The Simultaneous Chinese Remainder Problem appeared already under distinct forms in the literature. The upcoming examples are all based on pairwise coprime moduli and the traditional representative set $\mathcal{S}_M = \{0, \ldots, M - 1\}$ with $M = \prod_{i=1}^{k} m_i$. [BN00, Problem 3] introduced the *Noisy* Chinese Remainder Problem which essentially consists in a Simultaneous Chinese Remainder Problem with remainder sets of a fixed common size m with the objective of finding all sufficiently small solutions. This problem was a byproduct of their new polynomial reconstruction method. The series of papers [ZX97, Xia99, Xia00] considers the detection of frequencies in under-sampled waveforms, which can be reduced to a Simultaneous Chinese Remainder Problem where a set of particular solutions needs to be filtered out. [Lip09] introduced the Chinese Remainder Theorem with limits which corresponds to the Interval Simultaneous Chinese Remainder Problem.

11.4 Applications

Theorem 11.2 implies that an efficient Minimal Simultaneous Chinese Remainder Problem solver could be used to solve any NP problem. Thus, the Minimal Simultaneous Chinese Remainder Problem has a prestigious amount of applications. Besides those complexity-theoretic related consequences, some direct applications can be outlined.

11.4.1 Factoring with extra information

The first application consists in an elementary factoring algorithm. However, this factorisation requires some extra information about the factors.

Motivating example

We start with a quick look at what information is sufficient to deterministically filter out a factor of a given integer. To do so, let $N = p \cdot q \in \mathbb{N}$ be a composite integer $1 < p < q$. If for some pairwise coprime moduli $m_1, \dots, m_k$ such that $p < M = \prod_{i=1}^{k} m_i$, the remainders $p_i = [\![p_i \mod m_i]\!]$ are known, then the classical Chinese Remainder Theorem may be used to recover p and decompose N into its two factors. Often those remainders are not known exactly, but can be reduced to a small set of potential remainders; in other words, instead of knowing the remainders p_i, one only knows that $[\![p \mod m_i]\!] \in \mathcal{R}_i$ for some nonempty remainder set $\mathcal{R}_i \subset \{0, \dots, m_i - 1\}$. The problem of recovering p in plain becomes a Simultaneous Chinese Remainder Problem.

Remainder sets of size 2

A comparably easy problem instance may be obtained by granting only two possibilities for p_i (i.e. $|\mathcal{R}_i| = 2$) for all $i \in \{1, \dots, k\}$. In this case, Coppersmith's theorem (see Theorem 15.1) predicts that p can be found in time polynomial in $(\log(M), 2^2)$, provided that $p < \sqrt{M}$. This scenario may occur if for each modulus m_i, the remainder r_i of one factor of N is given but cannot be assigned to its specific factor such that either $r_i \equiv p \mod m_i$ or $r_i \equiv q \mod m_i$.

General remainder sets

Assume that for each m_i the remainder of p is known up to a small number of possibilities $[\![p \mod m_i]\!] \in \mathcal{R}_i$. Assume, in addition, that $p < M$. Then, p is a primitive solution of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k))$. If M, is sufficiently large, then p can be expected to be one of the smallest solutions, and so a Minimal Chinese Remainder Problem may find it.

Concrete application

In [BN00], the authors suggested the factorization of integers of the form $N = p^2 q$. More precisely, N is a quadratic residue modulo m_i if and only if q is so. Thus, q can be determined up to half of all possible remainders for each m_i.

11.4.2 Point counting on elliptic curves

The next application was described in [BN00] and consists in point counting on elliptic curves over finite fields. The main idea of this application is to use the *Schoof-Elkies-Atkins* (SEA) algorithm [Sch95], but instead of only using the good *Elkies primes* resulting in exact remainder information, also *Atkins primes* resulting in imprecise remainder information is used. This loss in precision is compensated by the screening of a Simultaneous Chinese Remainder Problem instance over the *Hasse* range [Has36]. We refer to [BN00] for a detailed development.

Chapter 12

An upper bound for the minimal solution

In Chapter 11 we focused on some variants of the Simultaneous Chinese Remainder Problem aiming at finding a solution with particular properties such as minimality, maximality, or being contained in a specified interval. Proposition 11.5 claims that all of these problems are polynomially equivalent and Theorem 11.2 yields that they are $\mathsf{NP}-\mathsf{complete}$. Yet not all instances are equally hard. In this chapter, we develop some elementary upper bounds for the minimal solution of a Simultaneous Chinese Remainder Problem. We note that by the polynomial reduction in Proposition 11.5, any such result also yields a lower bound for the maximal solution.

Hereinafter, we consider Simultaneous Chinese Remainder Problem instances of the form

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), \dots, (m_k, \mathcal{R}_k), \mathcal{S}_M) \tag{12.1}$$

with pairwise coprime moduli and the fixed representative set $\mathcal{S}_M = \{0, \dots, M-1\}$ with $M = \prod_{i=1}^{k} m_i$. As there is no ambiguity, we drop $\mathcal{S}_M$ from our notation. Furthermore, we focus on the Minimal Simultaneous Chinese Remainder Problem only.

12.1 Intuition

Going back to Garner's algorithm (Algorithm 8.2), we observe that any $x \in \{0, \dots, M-1\}$ can be uniquely represented as a sum

$$x = y_1 + m_1(y_2 + \cdots + m_{k-2}(y_{k-2} + m_{k-1}y_k) \dots) \tag{12.2}$$

where $y_i \in \{0, \ldots, m_i - 1\}$ for all $i \in \{1, \ldots, k\}$. Thereby, we obtain a bijection

$$\{0, \ldots, M-1\} \leftrightarrow \{0, \ldots, m_1 - 1\} \times \cdots \times \{0, \ldots, m_k - 1\}. \tag{12.3}$$

In general, this bijection does not correspond to the isomorphism from Theorem 8.7 as $y_i \neq [\![x \mod m_i]\!]$. Let f denote this bijection such that $f(x) = (y_1, \ldots, y_k)$ and let $x' \in \{0, \ldots, M-1\}$ be a second element with $f(x') = (y'_1, \ldots, y'_k)$. Equation (12.2) shows that $x < x'$ if and only if $y_i < y'_i$ for some $i \in \{1, \ldots, k\}$ and $y_j = y'_j$ for all $j \in \{1, \ldots, i-1\}$. This defines the lexicographic order on $\{0, \ldots, m_1 - 1\} \times \cdots \times \{0, \ldots, m_k - 1\}$, granting a mapping between comparison functions. This conclusion is used by so-called *mixed radix comparison* aiming at comparing elements in a *residue number system* [ST67]. Moreover, it yields an elementary upper bound B_0 for the maximal minimal solution of a Simultaneous Chinese Remainder Problem. Concretely, given the number of remainders in each remainder set, we can predict the maximal minimal solution over any choice of remainders with a fixed remainder set size, which is

$$B_0 - 1 := f(m_1 - |\mathcal{R}_1|, \ldots, m_k - |\mathcal{R}_k|). \tag{12.4}$$

Based on a simple counting argument on the maximal number of solutions inside a given interval, we manage to formally prove this conclusion and gain some additional insights.

12.2 Maximal number of solutions inside a given interval

We start by studying the maximal number of primitive solutions inside a given interval.

Lemma 12.1. *Let $m_1, \ldots, m_k \geq 2$ be pairwise coprime integers. For all $i \in \{1, \ldots, k\}$, let $\mathcal{R}_i \subseteq \{0, 1, \ldots, m_i - 1\}$ be a non-empty set of possible remainders modulo m_i and let $t_i = |\mathcal{R}_i|$ be its size. Furthermore, let $j \in \{1, \ldots, k\}$ and set $m_0 = t_0 = 1$. Then, any set $I \subseteq \mathbb{Z}$ consisting of $\mathfrak{M}_j := \prod_{i=0}^{j} m_i$ consecutive integers contains at most $\mathfrak{T}_j := \prod_{i=0}^{j} t_i$ solutions of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k))$.*

Proof. Assume by contradiction that for some $j \in \{1, \ldots, k\}$ there is a set I consisting of $\mathfrak{M}_j$ consecutive integers that contains $\mathfrak{T}_j + 1$ distinct primitive solutions of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k))$. Let $y = \min\{I\}$, $\overline{I} = \{x - y \mid x \in I\}$, and for all $i \in \{1, \ldots, k\}$ let

$$\overline{\mathcal{R}}_i = \{[\![r - y \mod m_i]\!] \mid r \in \mathcal{R}_i\}. \tag{12.5}$$

Then, $\overline{I} = \{0, \ldots, \mathfrak{M}_j - 1\}$ contains $\mathfrak{T}_j + 1$ distinct primitive solutions of $\mathsf{SimCRP}((m_1, \overline{\mathcal{R}}_1), \ldots, (m_k, \overline{\mathcal{R}}_k))$. Let $0 \le x_1 < \cdots < x_{\mathfrak{T}_j+1} < \mathfrak{M}_j$ denote these solutions. Then, for each $i \in \{1, \ldots, \mathfrak{T}_j + 1\}$, x_i is the solution of a Chinese Remainder Problem $\mathsf{CRP}((m_1, r_{1_i}), \ldots, (m_j, r_{j_i}))$ where $r_{s_i} \in \overline{\mathcal{R}}_s$ for all $s \in \{1, \ldots, j\}$. As the solutions are distinct and strictly smaller than $\mathfrak{M}_j$, any two of the $\mathfrak{T}_j + 1$ corresponding Chinese Remainder Problems differ in at least one congruence. However, there are only $\mathfrak{T}_j$ possibilities to form pairwise distinct Chinese Remainder Problems with $\mathcal{R}_1, \ldots, \mathcal{R}_k$. This yields the desired contradiction. □

12.3 Elementary upper bound

The rough approximation of the number of solutions in a fixed interval described in Lemma 12.1 is sufficient to deduce an unconditional upper bound for the maximal minimal solution of a Simultaneous Chinese Remainder Problem with fixed remainder set sizes.

Theorem 12.2. *Let $m_1, \ldots, m_k \ge 2$ be pairwise coprime integers. For all $i \in \{1, \ldots, k\}$, let $\mathcal{R}_i \subseteq \{0, 1, \ldots, m_i - 1\}$ be a non-empty set of possible remainders modulo m_i and let $t_i = |\mathcal{R}_i|$ be its size. Furthermore, set $m_0 = t_0 = t_{k+1} = 1$.*

1. *For any $j \in \{0, 1, \ldots, k\}$, there are at least $\mathfrak{T}_{k-j} := \prod_{i=0}^{k-j} t_i$ primitive solutions of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k))$ strictly smaller than*

$$B_{k-j} := \mathfrak{M}_k - \sum_{i=k-j}^{k} (t_{i+1} - 1)\mathfrak{M}_i$$

   where $\mathfrak{M}_i := \prod_{\ell=0}^{i} m_\ell$ for all $i \in \{0, \ldots, k\}$.

2. *The minimal solution χ_{min} of $\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k))$ satisfies*

$$\chi_{min} < B_0.$$

Proof. $\mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k))$ has exactly $\mathfrak{T}_k = \prod_{i=1}^{k} t_i$ primitive solutions. Thus, there are $\mathfrak{T}_k$ primitive solutions strictly smaller than

$B_k = \mathfrak{M}_k$. Next, assume that for some $j \in \{0, \dots, k-1\}$ there are at least $\mathfrak{T}_{k-j} = \prod_{i=0}^{k-j} t_i$ primitive solutions strictly smaller than

$$B_{k-j} = \mathfrak{M}_k - \sum_{i=k-j}^{k} (t_{i+1} - 1)\mathfrak{M}_i = \mathfrak{M}_k - \sum_{i=1}^{j} (t_{k-i+1} - 1)\mathfrak{M}_{k-i}. \tag{12.6}$$

If $t_{k-(j+1)+1} = 1$, then $\mathfrak{T}_{k-j} = \mathfrak{T}_{k-(j+1)}$ and $B_{k-j} = B_{k-(j+1)}$ which implies the claim for $j+1$. Thus, assume $t_{k-(j+1)+1} \geq 2$. By Lemma 12.1, each set consisting of $\mathfrak{M}_{k-(j+1)}$ consecutive integers contains at most $\mathfrak{T}_{k-(j+1)}$ solutions. In particular, each set of the form

$$\{B_{k-j} - (a+1)\mathfrak{M}_{k-(j+1)}, \dots, B_{k-j} - a\mathfrak{M}_{k-(j+1)} - 1\} \tag{12.7}$$

with $a \in \{0, \dots, t_{k-(j+1)+1} - 2\}$ contains at most $\mathfrak{T}_{k-(j+1)}$ solutions. Therefore, there are at most $(t_{k-(j+1)+1} - 1)\mathfrak{T}_{k-(j+1)}$ primitive solutions in the set

$$\bigcup_{a=0}^{t_{k-(j+1)+1}-2} \{B_{k-j} - (a+1)\mathfrak{M}_{k-(j+1)}, \dots, B_{k-j} - a\mathfrak{M}_{k-(j+1)} - 1\} \tag{12.8}$$

$$= \left\{B_{k-j} - (t_{k-(j+1)+1} - 1)\mathfrak{M}_{k-(j+1)}, \dots, B_{k-j} - 1\right\} \tag{12.9}$$

and consequently, there are at least

$$\mathfrak{T}_{k-j} - (t_{k-(j+1)+1} - 1)\mathfrak{T}_{k-(j+1)} = \mathfrak{T}_{k-(j+1)} \tag{12.10}$$

primitive solutions strictly smaller than

$$B_{k-j} - (t_{k-(j+1)+1} - 1)\mathfrak{M}_{k-(j+1)} = B_{k-(j+1)}. \tag{12.11}$$

By induction, we conclude that the first claim holds for each $j \in \{0, 1, \dots, k\}$. Setting $j = k$, we deduce that there is at least $\mathfrak{T}_0 = 1$ solution strictly smaller than B_0. □

As $t_i \geq 1$ for all $i \in \{0, 1, \dots, k+1\}$ and $\mathfrak{M}_\ell > 0$ for all $\ell \in \{0, \dots, k\}$, the bounds in Theorem 12.2 satisfy

$$0 \leq B_0 \leq B_1 \leq \dots \leq B_k = \mathfrak{M}_k \tag{12.12}$$

12.4 Optimal ordering

We highlight that the order of the moduli in Theorem 12.2 has an impact on the resulting upper bounds. The analysis of an optimal ordering of the moduli requires an auxiliary result.

Lemma 12.3. *Let $m, m' \geq 2$ be positive integers and $t, t' \in \mathbb{R}$. Then,*

$$\frac{t'-1}{m'} + \frac{t-1}{mm'} \leq \frac{t-1}{m} + \frac{t'-1}{mm'} \qquad \Leftrightarrow \qquad \frac{t'-1}{m'-1} \leq \frac{t-1}{m-1}.$$

Proof. We observe that:

$$\begin{aligned} \frac{t'-1}{m'} + \frac{t-1}{mm'} \leq \frac{t-1}{m} + \frac{t'-1}{mm'} \Leftrightarrow \quad & \frac{t'-1}{m'} - \frac{t'-1}{mm'} \leq \frac{t-1}{m} - \frac{t-1}{mm'} && (12.13) \\ \Leftrightarrow \quad & (t'-1)\frac{m-1}{m'm} \leq (t-1)\frac{m'-1}{mm'} && (12.14) \\ \Leftrightarrow \quad & \frac{t'-1}{m'-1} \leq \frac{t-1}{m-1}. && (12.15) \end{aligned}$$

□

Proposition 12.4. *Let $m_1, \dots, m_k \geq 2$ be pairwise coprime integers. For all $i \in \{1, \dots, k\}$, let $\mathcal{R}_i \subseteq \{0, 1, \dots, m_i - 1\}$ be a nonempty set of possible remainders modulo m_i and let $t_i = |\mathcal{R}_i|$ be its size. Furthermore, set $m_0 = t_{k+1} = 1$. Let $B_0 := \mathfrak{M}_k - \sum_{i=0}^{k}(t_{i+1} - 1)\mathfrak{M}_i$ where $\mathfrak{M}_\ell := \prod_{i=0}^{\ell} m_i$ for all $\ell \in \{0, \dots, k\}$. Let σ be a permutation on $\{0, 1, \dots, k+1\}$ fixing 0 and $k+1$, in other words, $\sigma(0) = 0$ and $\sigma(k+1) = k+1$. Let $B'_0 := \mathfrak{M}'_k - \sum_{i=0}^{k}(t_{\sigma(i+1)} - 1)\mathfrak{M}'_i$ where $\mathfrak{M}'_\ell := \prod_{i=0}^{\ell} m_{\sigma(i)}$ for all $\ell \in \{0, \dots, k\}$.*

1. *Let σ be an adjacent transposition switching j and $j+1$ for some $j \in \{1, \dots k-1\}$ (that is, $\sigma(i) = i$ for all $i \in \{0, \dots k+1\} \setminus \{j, j+1\}$ and $\sigma(j) = j+1$, $\sigma(j+1) = j$). Then $B'_0 > B_0$ if and only if $\frac{t_j - 1}{m_j - 1} < \frac{t_{j+1}-1}{m_{j+1}-1}$. Furthermore, $B'_0 = B_0$ if and only if $\frac{t_j-1}{m_j-1} = \frac{t_{j+1}-1}{m_{j+1}-1}$.*
2. *B_0 is minimal if and only if $\frac{t_1-1}{m_1-1} \leq \frac{t_2-1}{m_2-1} \leq \dots \leq \frac{t_k-1}{m_k-1}$.*

Proof. For the first claim, we note that

$$\begin{aligned} & B'_0 > B_0 && (12.16) \\ \Leftrightarrow & \mathfrak{M}'_k - \sum_{i=0}^{k}(t_{\sigma(i+1)} - 1)\mathfrak{M}'_i > \mathfrak{M}_k - \sum_{i=0}^{k}(t_{i+1} - 1)\mathfrak{M}_i && (12.17) \end{aligned}$$

$$\begin{aligned}\Leftrightarrow&(-t_{\sigma(j)}+1)\mathfrak{M}'_{j-1}+(-t_{\sigma(j+1)}+1)\mathfrak{M}'_j\\ &\qquad>(-t_j+1)\mathfrak{M}_{j-1}+(-t_{j+1}+1)\mathfrak{M}_j\end{aligned}\tag{12.18}$$

$$\begin{aligned}\Leftrightarrow&(-t_{j+1}+1)\mathfrak{M}_{j-1}+(-t_j+1)\mathfrak{M}_{j-1}m_{j+1}\\ &\qquad>(-t_j+1)\mathfrak{M}_{j-1}+(-t_{j+1}+1)\mathfrak{M}_j\end{aligned}\tag{12.19}$$

$$\Leftrightarrow(-t_{j+1}+1)+(-t_j+1)m_{j+1}>(-t_j+1)+(-t_{j+1}+1)m_j\tag{12.20}$$

$$\Leftrightarrow(-t_j+1)(m_{j+1}-1)>(-t_{j+1}+1)(m_j-1)\tag{12.21}$$

$$\Leftrightarrow(t_j-1)(m_{j+1}-1)<(t_{j+1}-1)(m_j-1)\tag{12.22}$$

$$\Leftrightarrow\frac{t_j-1}{m_j-1}<\frac{t_{j+1}-1}{m_{j+1}-1}\tag{12.23}$$

The equality statement is obtained by using in the above inequalities the equal sign.

For the second claim, assume first that B_0 is minimal and assume by contradiction that there are $\ell_1,\ell_2\in\{1,\dots k\}$ such that $\ell_1<\ell_2$ and $\frac{t_{\ell_2}-1}{m_{\ell_2}-1}<\frac{t_{\ell_1}-1}{m_{\ell_1}-1}$. Then, there exists $j\in\{\ell_1,\dots,\ell_2-1\}$ such that $\frac{t_{j+1}-1}{m_{j+1}-1}<\frac{t_j-1}{m_j-1}$. By the first claim $B'_0:=\mathfrak{M}'_k-\sum_{i=0}^k(t_{\sigma(i+1)}-1)\mathfrak{M}'_i$ where $\mathfrak{M}'_\ell:=\prod_{i=0}^\ell m_{\sigma(i)}$ and σ is the adjacent transposition switching j and $j+1$ is smaller than B_0 directly contradicting its minimality. Reciprocally, assume that $\frac{t_1-1}{m_1-1}\leq\frac{t_2-1}{m_2-1}\leq\cdots\leq\frac{t_k-1}{m_k-1}$ and assume by contradiction that B_0 is not minimal. There exists a minimal $B'_0:=\mathfrak{M}'_k-\sum_{i=0}^k(t_{\sigma(i+1)}-1)\mathfrak{M}'_i$ such that $B'_0<B_0$ for some permutation $\sigma\neq Id$. In particular, there exists $\ell_1,\ell_2\in\{1,\dots,k\}$ such that $\ell_1<\ell_2$, $\sigma(\ell_2)<\sigma(\ell_1)$, and $\frac{t_{\sigma(\ell_2)}-1}{m_{\sigma(\ell_2)}-1}<\frac{t_{\sigma(\ell_1)}-1}{m_{\sigma(\ell_1)}-1}$ directly contradicting our assumption $\frac{t_1-1}{m_1-1}\leq\frac{t_2-1}{m_2-1}\leq\cdots\leq\frac{t_k-1}{m_k-1}$.

□

12.5 Limitations

We note that the bound B_0 in Theorem 12.2 represents a bound on the maximal minimum, and, as such, it is not tight in general. Even its improvement from Proposition 12.4 is usually not achieved. Indeed, asymptotically, the bound B_0 is of size $O\left(\frac{M}{\max m_i}\right)$ where $M=\prod_{i=1}^k m_i$, but empirical observations predict that the minimal solution is magnitudes smaller (see Section 18.4).

Chapter 13

Deterministic solving method

By Theorem 11.2 we cannot expect to find a polynomial-time solving method of the Minimal Simultaneous Chinese Remainder Problem. Yet, it is interesting to study its solving methods. In this chapter, we start this study by proving that a solving method does not need to compute all solutions of a given Simultaneous Chinese Remainder Problem instance to find its minimum.

Let $m_1, ..., m_k \geq 2$ be pairwise coprime integers, for all $i \in \{1, ..., k\}$, let $\mathcal{R}_i = \{r_{i,1}, \ldots, r_{i,t_i}\} \subseteq \{0, 1, ..., m_i - 1\}$ be a non-empty set of remainders modulo m_i and consider the Simultaneous Chinese Remainder Problem $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$.

13.1 The trivial case: k=1

When facing a single modulus m_1, the Simultaneous Chinese Remainder Problem turns out to be a simple comparison problem. Indeed, given m_1 and $\mathcal{R}_1$, it is sufficient to find $\min_{r_1 \in \mathcal{R}_1} r_1$ to solve $\mathsf{SimCRP}((m_1, \mathcal{R}_1))$. This minimum may be found through any suitable comparison function and is linear in $|\mathcal{R}_1|$.

13.2 The first non-trivial case: k=2

The first nontrivial example of a Simultaneous Chinese Remainder instance arises when considering two moduli m_1, m_2. Indeed, due to the chaotic modular metric, a direct comparison of remainders does usually not yield the minimal solution. There are constructions of comparison functions in residue number systems [Isu16], but computing the minimum by means of such a function requires $|\mathcal{R}_1||\mathcal{R}_2|$ comparisons.

13.2.1 A solving algorithm

The question arises whether there is any method for finding the minimal solution in less than $|\mathcal{R}_1||\mathcal{R}_2|$ steps. We answer this question positively by putting forth the following algorithm inspired by Garner's algorithm (Algorithm 8.2).

Algorithm 13.1: Minimal Simultaneous Chinese Remaindering for two moduli.

```
Input: $\mathsf{SimCRP}((m_1, \mathcal{R}_1), (m_2, \mathcal{R}_2))$.
Output: The algorithm computes the minimal solution $\chi$.
$C_2 \leftarrow [\![ m_1^{-1} \mod m_2 ]\!]$
$\mathcal{R}_1^* \leftarrow \{ [\![ -r_1 C_2 \mod m_2 ]\!] \mid r_1 \in \mathcal{R}_1 \}$
$\mathcal{R}_2^* \leftarrow \{ [\![ r_2 C_2 \mod m_2 ]\!] \mid r_2 \in \mathcal{R}_2 \}$
$\mathcal{R}_1^* \leftarrow \mathrm{sort}(\mathcal{R}_1^*)$
$\mathcal{R}_2^* \leftarrow \mathrm{sort}(\mathcal{R}_2^*)$
$x_{-1} \leftarrow \min_{r_1^* \in \mathcal{R}_1^*} r_1^* + \min_{r_2^* \in \mathcal{R}_2^*} r_2^*$
for each $r_1^* \in \mathcal{R}_1^*$ do
    $x_{r_1^*} \leftarrow \min^+_{r_2^* \in \mathcal{R}_2^*} (r_1^* + r_2^* - m_2)$
$y_2 \leftarrow \min_{r_1^* \in \mathcal{R}_1^* \cup \{-1\}} x_{r_1^*}$
$\mathcal{R}^s \leftarrow \{ r_1^* \in \mathcal{R}_1^* \cup \{-1\} \mid x_{r_1^*} = y_2 \}$
$y_1 \leftarrow \min_{r_1^* \in \mathcal{R}^s} [\![ -r_1^* m_1 \mod m_2 ]\!]$
return $\chi \leftarrow y_1 + m_1 y_2$
```

13.2.2 Analysis

On input (m_1, r_1), (m_2, r_2) Garner's algorithm computes $C_1 := 1$, $C_2 := [\![m_1^{-1} \mod m_2]\!]$, $y_1 := r_1$, $y_2 := [\![(r_2 - r_1) C_2 \mod m_2]\!]$ and outputs the solution $\chi = y_1 + m_1 y_2$. To find the minimal value when varying the remainders in the input, we need to minimize

$$
\begin{aligned}
y_2 &= [\![(r_2 - r_1) C_2 \mod m_2]\!] && (13.1)\\
&= [\![[\![r_2 C_2 \mod m_2]\!] + [\![-r_1 C_2 \mod m_2]\!] \mod m_2]\!] \,. && (13.2)
\end{aligned}
$$

To do so, we first compute

$$\mathcal{R}_1^* = \{[\![-r_1 C_2 \mod m_2]\!] \mid r \in \mathcal{R}_1\}, \text{ and} \tag{13.3}$$

$$\mathcal{R}_2^* = \{[\![r_2 C_2 \mod m_2]\!] \mid r_2 \in \mathcal{R}_2\}, \tag{13.4}$$

which requires $|\mathcal{R}_1| + |\mathcal{R}_2|$ modular multiplications. Next, we sort the elements of $\mathcal{R}_1^*$ and $\mathcal{R}_2^*$ in increasing order, which requires $|\mathcal{R}_i| \log(|\mathcal{R}_i|)$ comparisons for each $i \in \{1, 2\}$. Finally, we need to combine the elements from those two sets to find the minimal value modulo m_2. As exactly one element from each set needs to be chosen and as each such element is non-negative and strictly smaller than m_2, the modulo operation is either not carried out at all or m_2 is subtracted once. Thus, the minimum either corresponds to the sum of the minima of both sets, or it corresponds to the minimal sum greater than m_2. With the elements being ordered, we can efficiently find those combinations. Indeed, for the sum of the minima of both sets, we only need to sum the first element in each list. For a minimal sum larger than m_2, we go through all the elements $r_1^* \in \mathcal{R}_1^*$ and find the minimizing element $r_2^* \in \mathcal{R}_2^*$. This minimizing element can be found in $O(|\mathcal{R}_2^*| \log(|\mathcal{R}_2^*|))$ steps through a logarithmic search. The sum of r_1^* and the minimizing element in $\mathcal{R}_2^*$ is reduced modulo m_2 and stored as $x_{r_1^*}$. The minimum of these values is set to be y_2. Subsequently, the corresponding minimal value for y_1 is defined by the minimal remainder r_1 whose attributed r_1^* allows us to achieve y_2. All in all, the algorithm only requires $O(t \log(t))$ steps where $t = \max(|\mathcal{R}_1|, |\mathcal{R}_2|)$.

Remark 13.1. *Algorithm 13.1 consists in a time-space trade-off improving the efficiency of a Minimal Simultaneous Chinese Remainder solver. Whereas a direct comparison requires $O(t^2)$ steps to find the minimum, it does not need to store any additional variables. Algorithm 13.1 stores auxiliary values in $\mathcal{R}_1^*, \mathcal{R}_2^*, \mathcal{R}^s$, but improves the number of required steps to $O(t \log(t))$.*
Especially for large moduli sets, a substantial gain is obtained.

13.3 The general case

In the general case, we may recycle the previous idea to simplify the overall complexity. Given $m_1, ..., m_k$ and corresponding remainder sets $\mathcal{R}_1, \ldots, \mathcal{R}_k$, computing all Simultaneous Chinese Remainder solutions and comparing them to find the minimum requires $O(\prod_{i=1}^k |\mathcal{R}_i|)$ steps. In particular, if all the remainder sets are of approximately the same size, then $O(t^k)$ steps are required where $t = \max\{|\mathcal{R}_1|, \ldots, |\mathcal{R}_k|\}$.

On the other hand, we can simplify this computation by splitting the remainder sets into two disjoint groups $\mathcal{R}^{(1)}$ and $\mathcal{R}^{(2)}$ such that

$$\prod_{\mathcal{R}_i \in \mathcal{R}^{(1)}} |\mathcal{R}_i| \approx \prod_{\mathcal{R}_j \in \mathcal{R}^{(2)}} |\mathcal{R}_j|. \tag{13.5}$$

In this way, we can compute the solution to all Chinese Remainder Problem instances from each group using traditional Chinese Remaindering and subsequently use Algorithm 13.1 to recombine the resulting groups. The number of steps shrinks from $O(t^k)$ to $O(kt^{k/2}\log(t))$.

Remark 13.2. *Any state-of-the-art solving method of the Minimal Simultaneous Chinese Remainder Problem can be expected to have an exponential runtime. Indeed, the currently best 3-SAT solvers run in time* 1.307^n *[HKZZ19]. As by Theorem 11.2 the Minimal Simultaneous Chinese Remainder Problem is* $\mathsf{NP}-\mathsf{hard}$, *its solving methods cannot be expected to be substantially faster.*

Chapter 14

Family of minimizing functions

Before we continue to study solving methods of the Minimal Simultaneous Chinese Remainder Problem, we outline a peculiar result on mixed radix comparison. This result allows us to construct a family of functions which minimize in either the minimal or maximal solution of a Simultaneous Chinese Remainder Problem instance. Our development consists in a direct generalization of [DIP93].

14.1 Declaring a new modulus and auxiliary coefficients

Hereinafter, let $m_1, \dots, m_k$ be pairwise coprime moduli, let $m_{\max} = \max_{i \in \{1,\dots,k\}} m_i$, set $M = \prod_{i=1}^{k} m_i$, and $M_i = \frac{M}{m_i}$ for all $i \in \{1, \dots, k\}$. We define a new modulus

$$SQ = c_1 M_1 + \cdots + c_k M_k \tag{14.1}$$

where $c_1, \dots, c_k \in \mathbb{N}$ such that $\gcd(c_i, m_i) = 1$ for all $i \in \{1, \dots, k\}$. This new modulus is pairwise coprime to all the initial moduli.

Lemma 14.1. *For all* $i \in \{1, \dots, k\}$, *we have* $\gcd(SQ, m_i) = 1$.

Proof. By construction, we have

$$\gcd(SQ, m_i) = \gcd(c_1 M_1 + \cdots + c_k M_k, m_i) = \gcd(c_i M_i, m_i) = 1 \tag{14.2}$$

□

Coprimality between SQ and m_i allows us to compute the inverse of m_i modulo SQ and to define the additional variables

$$\kappa_i = [\![-c_i m_i^{-1} \mod SQ]\!] \tag{14.3}$$

for all $i \in \{1, \ldots, k\}$. A remarkable property of those κ_i is that their collective sum is a multiple of SQ.

Lemma 14.2. *Using the above definition, we have $\kappa_1 + \cdots + \kappa_k \equiv 0 \mod SQ$.*

Proof. We have

$$\begin{aligned}
&\kappa_1 + \cdots + \kappa_k \equiv 0 \mod SQ && (14.4)\\
\Leftrightarrow &- M(\kappa_1 + \cdots + \kappa_k) \equiv 0 \mod SQ && (14.5)\\
\Leftrightarrow &- \kappa_1 m_1 M_1 - \cdots - \kappa_k m_k M_k \equiv 0 \mod SQ && (14.6)\\
\Leftrightarrow &c_1 M_1 + \cdots + c_k M_k \equiv 0 \mod SQ && (14.7)
\end{aligned}$$

where the last congruence holds by the definition of SQ. □

14.2 Constructing new functions minimizing in an extremum

We construct a particular comparison function over $\{0, 1, \ldots, M-1\}$.

Proposition 14.3. *The function*

$$D^+ : \{0, 1, \ldots, M-1\} \to \{0, \ldots, SQ-1\}$$

defined by $D^+(X) = [\![\kappa_1 x_1 + \cdots + \kappa_k x_k \mod SQ]\!]$ where $x_i = [\![X \mod m_i]\!]$ for all $i \in \{1, \ldots, k\}$ is non-decreasing on $\{0, 1, \ldots, M-1\}$.

Proof. As for each $X \in \{0, 1, \ldots, M-1\}$ and each $i \in \{1, \ldots, k\}$, we have $x_i = [\![X \mod m_i]\!] = \left(X - \left\lfloor \frac{X}{m_i} \right\rfloor m_i\right)$, we deduce that

$$\begin{aligned}
&D^+(X) && (14.8)\\
&= \left[\!\!\left[\kappa_1 \left(X - \left\lfloor \frac{X}{m_1} \right\rfloor m_1\right) + \cdots + \kappa_k \left(X - \left\lfloor \frac{X}{m_k} \right\rfloor m_k\right) \mod SQ\right]\!\!\right] && (14.9)\\
&= \left[\!\!\left[X(\kappa_1 + \cdots + \kappa_k) - \kappa_1 \left\lfloor \frac{X}{m_1} \right\rfloor m_1 - \cdots - \kappa_k \left\lfloor \frac{X}{m_k} \right\rfloor m_k \mod SQ\right]\!\!\right]. && (14.10)
\end{aligned}$$

By Lemma 14.2, $(\kappa_1 + \cdots + \kappa_k) \equiv 0 \mod SQ$ and by the definition of κ_i, $\kappa_i m_i \equiv -c_i \mod SQ$ for all $i \in \{1, \ldots, k\}$. Thus,

$$D^+(X) = \left\llbracket c_1 \left\lfloor \frac{X}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{X}{m_k} \right\rfloor \qquad \mod SQ \right\rrbracket . \tag{14.11}$$

As $X < M$, also $\left\lfloor \frac{X}{m_i} \right\rfloor < M_i$ for all $i \in \{1, \ldots, k\}$ implying

$$c_1 \left\lfloor \frac{X}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{X}{m_k} \right\rfloor < SQ \tag{14.12}$$

and so

$$D^+(X) = c_1 \left\lfloor \frac{X}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{X}{m_k} \right\rfloor \tag{14.13}$$

over the integers. As each floor function is non-decreasing in X, so is D^+. □

Remark 14.4. *We remark that for $X < Y \in \{0, \ldots, M-1\}$, we either have $D^+(X) < D^+(Y)$, or $x_i < y_i$ for all $i \in \{1, \ldots, k\}$. Thus, D^+ consists in a mixed radix comparison function. [DIP93] obtained this conclusion for $c_1 = \cdots = c_k = 1$.*

Corollary 14.5. *Let $X_1, \ldots, X_z \in \{0, \ldots, M-1\}$ for some $z \in \mathbb{Z}_{\geq 1}$, then*

$$\min_{i \in \{1,\ldots,z\}} D^+(X_i) = D^+\left(\min_{i \in \{1,\ldots,z\}} X_i \right) .$$

Thus, D^+ offers an elegant way to find the minimum among a given set of elements. However, often the minimum in absolute value is required. Therefore, we construct the same function over $\mathbb{Z}_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right]$.

Proposition 14.6. *The function*

$$D : \mathbb{Z}_M \to \mathbb{Z}_{SQ}$$

defined by $D(X) = [\kappa_1 x_1 + \cdots + \kappa_k x_k \mod SQ]$ where $x_i = \llbracket X \mod m_i \rrbracket$ for all $i \in \{1, \ldots, k\}$ is non-decreasing on $\mathbb{Z} \cap \left(-\frac{M}{2} + m_{\max}, \frac{M}{2}\right]$. Furthermore, $D(X) \geq 0$ if and only if $X \geq 0$.

Proof. Following the proof of Proposition 14.3, we deduce that

$$D(X) = \left[c_1 \left\lfloor \frac{X}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{X}{m_k} \right\rfloor \qquad \mod SQ \right] . \tag{14.14}$$

As $-\frac{M}{2}+m_{\max} < X \leq \frac{M}{2}$, we have $-\frac{M_i}{2} < \left\lfloor -\frac{M_i}{2}+\frac{m_{\max}}{m_i}\right\rfloor \leq \left\lfloor \frac{X}{m_i}\right\rfloor \leq \frac{M_i}{2}$ for all $i \in \{1,\dots,k\}$ implying

$$-\frac{SQ}{2} < c_1\left\lfloor\frac{X}{m_1}\right\rfloor + \cdots + c_k\left\lfloor\frac{X}{m_k}\right\rfloor \leq \frac{SQ}{2} \tag{14.15}$$

and so

$$D(X) = c_1\left\lfloor\frac{X}{m_1}\right\rfloor + \cdots + c_k\left\lfloor\frac{X}{m_k}\right\rfloor \tag{14.16}$$

over the integers. As each floor function is non-decreasing in X, so is D. Furthermore, if $Y \in \mathbb{R}$, then $\lfloor Y\rfloor \geq 0$ if and only if $Y \geq 0$. Thus, $D(X) \geq 0$ if and only if $X \geq 0$. □

Remark 14.7. *For $X \in \mathbb{Z}\cap\left(-\frac{M}{2}, -\frac{M}{2}+m_{\max}\right]$, we have*

$$-\frac{SQ}{2} - \sum_{i=1}^{k}\frac{c_i}{2} \leq c_1\left\lfloor\frac{X}{m_1}\right\rfloor + \cdots + c_k\left\lfloor\frac{X}{m_k}\right\rfloor \leq -\frac{SQ}{2} + \sum_{i=1}^{k} c_i\frac{m_{\max}}{m_i} \tag{14.17}$$

and so, if the middle sum is smaller than or equal to $-\frac{SQ}{2}$, a modular reduction is peformed by $D(X)$. In this case, the resulting integer is at least of size $\frac{SQ}{2}-\sum_{i=1}^{k}\frac{c_i}{2}$. To guarantee monotonicity of D, we need to eliminate those values from the considered set.

Corollary 14.8. *Let $\{X_1,\dots,X_z\} \subseteq \mathbb{Z}\cap\left(-\frac{M}{2}+m_{\max}, \frac{M}{2}\right]$ for some $\mathbb{Z}_{\geq 1}$. Let $\overline{X} \in \{X_1,\dots,X_z\}$ be such that:*

1. *Function minimality: $|D(\overline{X})| = \min_{i\in\{1,\dots,z\}}|D(X_i)|$, and*

2. *Representation minimality: for all $X' \in \{X_1,\dots,X_z\}$ that satisfies $|D(X')| = \min_{i\in\{1,\dots,z\}}|D(X_i)|$ we have $|\overline{X}| \leq |X'|$.*

Then, $\overline{X}$ is either the minimal non-negative element or the maximal negative element in $\{X_1,\dots,X_z\}$.

Proof. Let

$$X^{+}_{\min} = \min\{X_i \geq 0 \mid i \in \{1,\dots,z\}\}, \tag{14.18}$$

$$X^{-}_{\max} = \max\{X_i < 0 \mid i \in \{1,\dots,z\}\}. \tag{14.19}$$

Assume by contradiction that $\overline{X} \notin \{X^{+}_{\min}, X^{-}_{\max}\}$. If $\overline{X} \geq 0$, then, by minimality of $X^{+}_{\min}$, we have $\overline{X} > X^{+}_{\min}$ and by monotonicity of D, we have

$$0 \leq D(X^{+}_{\min}) \leq D(\overline{X}). \tag{14.20}$$

If the second inequality is strict, we get a contradiction with the function minimality of $\overline{X}$ and equality contradicts the representation minimality of $\overline{X}$. Thus, in this case, $\overline{X} = X^+_{\text{min}}$. Similarly, if $\overline{X} < 0$, then, by maximality of X^-_{max}, we have $\overline{X} < X^-_{\text{max}}$ and by monotonicity of D, we have

$$D(\overline{X}) \leq D(X^-_{\text{max}}) < 0 \tag{14.21}$$

and so

$$|D(\overline{X})| \geq |D(X^-_{\text{max}})| > 0 \tag{14.22}$$

If the first inequality in Equation (14.22) is strict, we get a contradiction with the function minimality of $\overline{X}$ and equality contradicts the representation minimality of $\overline{X}$. Thus, in this case, $\overline{X} = X^-_{\text{max}}$ proving the claim. □

Corollary 14.8 shows that the minimal function output of D in absolute value is either obtained for the minimal non-negative or the maximal negative element in a given set of comparison values. However, the minimal function output may not be reached by the minimal function input in absolute values. To better understand this bias, we note that for all $Y \in \mathbb{R}_+ \setminus \mathbb{Z}$, we have

$$\lfloor Y \rfloor < Y < |\lfloor -Y \rfloor| = \lfloor Y \rfloor + 1 \tag{14.23}$$

and for all $Y \in \mathbb{Z}_{\geq 0}$, we have

$$\lfloor Y \rfloor = Y = |\lfloor -Y \rfloor| < \lfloor Y \rfloor + 1. \tag{14.24}$$

Thus, for all $X \in \mathbb{Z} \cap \left(0, \frac{M}{2} - m_{\text{max}}\right)$, we have

$$\begin{aligned} D(X) &= c_1 \left\lfloor \frac{X}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{X}{m_k} \right\rfloor & (14.25)\\ &< c_1 \left| \left\lfloor \frac{-X}{m_1} \right\rfloor \right| + \cdots + c_k \left| \left\lfloor \frac{-X}{m_k} \right\rfloor \right| & (14.26)\\ &= \left| c_1 \left\lfloor \frac{-X}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{-X}{m_k} \right\rfloor \right| & (14.27)\\ &= |D(-X)| & (14.28) \end{aligned}$$

where the inequality in Equation (14.26) follows from the fact that $X \in \mathbb{Z} \cap \left(0, \frac{M}{2} - m_{\text{max}}\right)$ is not divisible by all moduli and the equality in Equation (14.27) is obtained as $c_i > 0$ for all $i \in \{1, \ldots, k\}$. Thereby, the function evaluation of a negative integer is in absolute value always larger than the function evaluation of its positive equivalent. Nonetheless, as

$$|D(-X)| = c_1 \left| \left\lfloor \frac{-X}{m_1} \right\rfloor \right| + \cdots + c_k \left| \left\lfloor \frac{-X}{m_k} \right\rfloor \right| \tag{14.29}$$

$$\leq c_1 \left(\left\lfloor \frac{X}{m_1} \right\rfloor + 1 \right) + \cdots + c_k \left(\left\lfloor \frac{X}{m_k} \right\rfloor + 1 \right) \tag{14.30}$$

$$= c_1 \left\lfloor \frac{X}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{X}{m_k} \right\rfloor + \sum_{i=1}^{k} c_i \tag{14.31}$$

$$= D(X) + \sum_{i=1}^{k} c_i, \tag{14.32}$$

the function evaluation of a negative integer is in absolute value bounded by the function evaluation of its positive equivalent.

14.3 Intermediate sizes

For efficiency reasons, one may ask about the size of the intermediate values and the final output of the function D^+ or D. Of course, intermediate values can be bounded as a function of SQ which itself is a function of $c_1, \ldots, c_k$ and $M_1, \ldots, M_k$. Very roughly, one can upper bound SQ by $CM_{\max}$ where $C = \sum_{i=1}^{k} c_i$ and $M_{\max} = \max_{i \in \{1,\ldots,k\}} M_i$. Assuming that c_i is polynomial in M, the new modulus is polynomial in M. The function outputs underlie the same comment. For example, for all $\mathbb{Z} \cap \left(-\frac{M}{2} + m_{\max}, \frac{M}{2}\right]$

$$|D(X)| \leq D(|X|) + \sum_{i=1}^{k} c_i, \tag{14.33}$$

$$\leq c_1 \left\lfloor \frac{|X|}{m_1} \right\rfloor + \cdots + c_k \left\lfloor \frac{|X|}{m_k} \right\rfloor + \sum_{i=1}^{k} c_i \tag{14.34}$$

$$\leq c_1 \frac{|X|}{m_1} + \cdots + c_k \frac{|X|}{m_k} + \sum_{i=1}^{k} c_i \tag{14.35}$$

$$= |X| \frac{c_1 M_1 + \cdots + c_k M_k}{M} + \sum_{i=1}^{k} c_i \tag{14.36}$$

$$= |X| \frac{SQ}{M} + C \tag{14.37}$$

14.4 A family of functions minimized in an extremum

The function D is completely defined by

$$D(X) = [\kappa_1 x_1 + \ldots, \kappa_k x_k \mod SQ]. \tag{14.38}$$

This implies that we can also work with the remainders

$$(x_1, \ldots, x_k) \in \mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_k} \tag{14.39}$$

instead of the recombined elements in $\mathbb{Z}_M$ to retrieve a minimizing element. Thus, modifying the underlying coefficients c_i of the function D results in a family of non-related functions defined by

$$SQ^\gamma = c_1^\gamma M_1 + \cdots + c_k^\gamma M_k \tag{14.40}$$

and

$$D^\gamma(X) = \left[\kappa_1^\gamma x_1 + \ldots, \kappa_k^\gamma x_k \mod SQ^\gamma\right] \tag{14.41}$$

each minimizing in either the maximal negative or the minimal non-negative element of a given set. In the next chapter, this conclusion is used to investigate the Simultaneous Chinese Remainder Problem from a new perspective.

Chapter 15

Lattice solving methods

In this chapter, we present two lattice-based solving methods to find the minimal primitive solution to the Simultaneous Chinese Remainder Problem. Both methods were already studied in [BN00] but we generalize and improve the second method.

Hereinafter, let $m_1, ..., m_k \geq 2$ be pairwise coprime integers and for all $i \in \{1, ..., n\}$ let $\mathcal{R}_i = \{r_{i,1}, \dots, r_{i,t_i}\} \subseteq \{0, 1, ..., m_i - 1\}$ be a non-empty set of possible remainders modulo m_i. Let $M = \prod_{i=1}^{k} m_i$ and $M_i = \frac{M}{m_i}$ for all $i \in \{1, \dots, k\}$. We consider the Simultaneous Chinese Remainder Problem instance

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k)). \tag{15.1}$$

Let χ denote any solution of this problem instance and let χ_{min} denote its minimal solution.

15.1 Coppersmith's method

The first solving technique is based on *Coppersmith*'s theorem on finding small roots of small degree univariate modular polynomials.

Theorem 15.1 ([Cop97, Corollary 1])**.** *Let $\mathcal{P}$ be a monic polynomial of degree T in one variable modulo $M \in \mathbb{Z}_{\geq 2}$. If $B \leq M^{\frac{1}{T}}$, then in time polynomial in $(\log(M), 2^T)$, we can find all $x_0 \in \mathbb{Z}$ such that $\mathcal{P}(x_0) \equiv 0 \mod M$ and $|x_0| \leq B$.*

To find these solutions, Coppersmith relies on three main observations. First, finding roots of real polynomials is efficient as shows for example the *Vincent-Collins-Akritas method* [RZ04]. Thus, if the modular polynomial can be transformed into an integer polynomial, it can also profit from those efficient algorithms. Second, if the coefficients of a modular polynomial are sufficiently small, then the corresponding real polynomial has the same roots. Third, a polynomial with large coefficients can be transformed into a polynomial with small coefficients through the construction of a particular lattice and the use of elementary lattice reduction (i.e., LLL). We skip the details of this construction and refer to [Gal12, Chapter 19] for an illustrative summary.

15.1.1 The approach

If the minimal solution of the considered Simultaneous Chinese Remainder Problem in Equation (15.1) is sufficiently small, then it suffices to construct first a polynomial whose roots correspond to the solutions of the Simultaneous Chinese Remainder Problem and subsequently use Coppersmith's result. The relation between Chinese Remaindering and Polynomial Interpolation described in Section 8.6 helps us with this construction. Indeed, let $T = \max_{i\in\{1,...,n\}}\{t_i\}$ and let for each $i \in \{1, .., n\}$, let

$$P_i(x) := (x - r_{i,1})^{T-t_i} \prod_{j=1}^{t_i} (x - r_{i,j}). \tag{15.2}$$

Then, for any solution χ of the Simultaneous Chinese Remainder Problem instance in Equation (15.1)

$$P_i(\chi) \equiv 0 \mod m_i. \tag{15.3}$$

Set

$$b_i := \left[\!\left[M_i^{-1} \mod m_i \right]\!\right] M_i \tag{15.4}$$

and compute

$$P(x) := \sum_{i=1}^{k} b_i P_i(x). \tag{15.5}$$

Then, $b_i \equiv \delta_{i,j} \mod m_j$, where $\delta_{i,j}$ denotes the Kronecker symbol defined by $\delta_{i,j} = 0$ whenever $j \neq i$ and $\delta_{i,i} = 1$. Thus, $P \equiv P_i \mod m_i$ for all $i \in \{1, ..., n\}$ which implies that:

1. $P(\chi) \equiv 0 \mod M$ as $P(\chi) \equiv P_i(\chi) \equiv 0 \mod m_i$ for all $i \in \{1, ..., k\}$.

2. $P(x)$ is of degree T as all the P_i's are so and the leading terms do not cancel out because $b_i > 0$ for all $i \in \{1, \ldots, k\}$.

3. $P(x)$ is monic modulo M as all the P_i's are monic, of the same degree and multiplied by coefficients that satisfy $\sum_{i=1}^{k} b_i \equiv 1 \mod M$.

Thereby, provided that $\chi_{\min}$ is sufficiently small, we can use Coppersmith's method for finding it.

Proposition 15.2. *If $\chi_{\min} < M^{\frac{1}{T}}$, then the minimal solution of the Simultaneous Chinese Remainder Problem* $\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k))$ *can be found in time polynomial in* $(\log(M), 2^T)$.

15.1.2 Comments

We note that the definition of the initial polynomials P_i in Equation (15.2) sets the first term in the product to be $(x - r_{i,1})^{T-t_i}$. This choice has the sole purpose of obtaining a degree T polynomial whose roots do not differ from the remainders $r_{i,1}, \ldots, r_{i,t_i}$. However, one may replace this auxiliary term with any other product of monomials whose roots do not differ from the given remainders and result in a degree $T - t_i$ polynomial.

We remark that the success of Coppersmith's method strongly depends on the size of the largest remainder set. Indeed, the largest remainder set defines T and so the upper bound $M^{\frac{1}{T}}$ for the minimal solution $\chi_{\min}$. A single large remainder set is sufficient to make the method impractical.

We need to pay attention to the fact that a solution of the considered Simultaneous Chinese Remainder Problem obtained by Coppersmith's method may not correspond to the minimal, but the maximal solution. Therefore, the sign needs to be checked.

Finally, we observe that the recursive use of the method allows us to always find the minimal solution. Indeed, if the minimal solution was not found in the interval $[0, M^{\frac{1}{T}}]$, then, the minimal primitive solution is greater than $M^{\frac{1}{T}}$. Thus, we can shift the solutions of the considered Simultaneous Chinese Remainder Problem instance in Equation (15.1) collectively by $2M^{\frac{1}{T}}$ to the left by computing

$$\widetilde{\mathcal{R}}_i = \left\{ \left[\!\left[r_i - \left\lfloor 2M^{\frac{1}{T}} \right\rfloor \mod m_i \right]\!\right] \;\middle|\; r \in \mathcal{R}_i \right\}, \tag{15.6}$$

and repeat the process with $\widetilde{\mathcal{R}}_i$. Now, either the minimal solution is found or we continue with another shift.

15.2 Bleichenbacher-Nguyen method

The second solving method represents the Simultaneous Chinese Remainder Problem as a particular sum over binary unknowns and mimics Lagrange's interpolation method to solve it.

15.2.1 First approach

Let

$$b_i := \left[M_i^{-1} \mod m_i\right] M_i \tag{15.7}$$

such that $b_i \equiv \delta_{i,j} \mod m_j$. Then, the minimal solution $\chi_{\min}$ modulo $M = \prod_{i=1}^{k} m_i$ satisfies

$$\chi_{\min} \equiv \sum_{i=1}^{k} \llbracket \chi_{\min} \mod m_i \rrbracket \, b_i \mod M \tag{15.8}$$

indicating that we only need to find the right indices of the corresponding remainders. To do so, we define unknown binary values $\eta_{i,j}$ by $\eta_{i,j} = 1$ if $\chi_{\min} \equiv r_{i,j} \mod m_i$ and $\eta_{i,j} = 0$ otherwise. With this notation, Equation (15.8) takes the form

$$\chi_{\min} \equiv \sum_{i=1}^{k} \sum_{j=1}^{t_i} \eta_{i,j} r_{i,j} b_i \mod M. \tag{15.9}$$

To find $\chi_{\min}$, set $T_\Sigma = \sum_{i=1}^{k} t_i$ and consider the $(T_\Sigma + 1) \times (T_\Sigma + 1)$ row lattice

$$\Lambda = \mathcal{L} \begin{pmatrix} M & 0 & 0 & \ldots & 0 \\ r_{1,1}b_1 & B & 0 & \ldots & 0 \\ r_{1,2}b_1 & 0 & B & \ldots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ r_{n,t_k}b_k & 0 & 0 & \ldots & B \end{pmatrix}, \tag{15.10}$$

whose elements are given by $(X, \eta_{1,1}B, \eta_{1,2}B, ..., \eta_{n,t_k}B) \in \mathbb{Z}^{T_\Sigma+1}$ such that $X \equiv \sum_{i=1}^{k} \sum_{j=1}^{m} \eta_{i,j} r_{i,j} b_i \mod M$. As desired, Λ contains the target vector

$$v := (\chi, \eta_{1,1}B, \eta_{1,2}B, ..., \eta_{n,t_k}B). \tag{15.11}$$

A quick analysis shows that the target vector has norm

$$\sqrt{\chi^2 + kB^2} \leq B\sqrt{k+1} \tag{15.12}$$

and is thus comparably short. Indeed, the determinant of Λ is $\det(\Lambda) = MB^{T_\Sigma}$ and so the Gaussian heuristic predicts a shortest vector of length

$$\lambda_1(\Lambda) \simeq \sqrt{\frac{T_\Sigma + 1}{2\pi e}} \det(\Lambda)^{\frac{1}{T_\Sigma+1}} = \sqrt{\frac{T_\Sigma + 1}{2\pi e}} (MB^{T_\Sigma})^{\frac{1}{T_\Sigma+1}}. \tag{15.13}$$

Thus, if $B\sqrt{k+1} \leq \sqrt{\frac{T_\Sigma+1}{2\pi e}}(MB^{T_\Sigma})^{\frac{1}{T_\Sigma+1}}$, then the target vector has a high chance of being a shortest vector.

Remark 15.3. *The Gaussian heuristic needs to be considered with caution. In [BN00, Appendix C], the authors remark that any sufficiently small linear combination of $r_{i,1}, ..., r_{i,t_i}$ gives rise to a shorter lattice point.*

To limit the possibility of bad recombinations, a block condition on the indices can be inserted, namely that

$$\sum_{j=1}^{t_{i_1}} \eta_{i_1,j} = \sum_{j=1}^{t_{i_2}} \eta_{i_2,j} \tag{15.14}$$

for all $1 \leq i_1, i_2 \leq k$. A lattice $\widetilde{\Lambda}$ with this additional condition can be obtained in polynomial time as the intersection of the full-dimensional lattice $\Lambda \subseteq \mathbb{Z}^{T_\Sigma+1}$ with the $(T_\Sigma - k + 1)$-dimensional vector subspace satisfying the above condition. Thus, $\widetilde{\Lambda}$ is a $(T_\Sigma - k + 1)$-dimensional lattice in $\mathbb{Z}^{T_\Sigma+1}$. Despite this improvement, bad cross combinations persist and the combination of two suitable basis vectors of $\widetilde{\Lambda}$ is sufficient to generate a shorter vector than v.

15.2.2 Remarks

The Bleichenbacher-Nguyen method enjoys a straightforward classic construction and it should find the minimal solution beyond Coppersmith's bound. However, it suffers from many undesired cross-combinations resulting in shorter lattice vectors than the target vector. Even after intersecting with a suitable vector space, most of those cross combinations remain. For example, the combination of the first, second, and third row in Equation (15.10) leads to

$$([(r_{1,1} - r_{1,2})b_1 \mod M], B, -B, 0, \ldots, 0)\,. \tag{15.15}$$

having only two entries equal to B in absolute value. In comparison, our target vector v in Equation (15.11) has k entries equal to B in absolute value. Thus, if $[(r_{1,1} - r_{1,2})b_1 \mod M]$ is sufficently small or B sufficiently large, then v is not be a shortest vector.

15.2.3 Second approach

To improve the Bleichenbacher-Nguyen method, we make use of the family of functions constructed in Chapter 14. Indeed, let $\Gamma \in \mathbb{N}_{\geq 1}$ and define, for each $\gamma \in \{1, \ldots, \Gamma\}$, a functional modulus

$$SQ^{(\gamma)} = c_1^{(\gamma)} M_1 + \cdots + c_k^{(\gamma)} M_k \tag{15.16}$$

for some $c_1^{(\gamma)}, \ldots, c_k^{(\gamma)} \in \mathbb{N}$ such that $\gcd(c_i^{(\gamma)}, m_i) = 1$ for all $i \in \{1, \ldots, k\}$. Define the corresponding functional coefficients as

$$k_i^{(\gamma)} := \left[-c_i^{(\gamma)} m_i^{-1} \mod SQ^{(\gamma)} \right] \tag{15.17}$$

for all $\gamma \in \{1, \ldots, \Gamma\}$ and all $i \in \{1, \ldots, k\}$. By Corollary 14.8, the functions $D^{(1)}, \ldots, D^{(\Gamma)}$ defined by these moduli are minimized for either the minimal or maximal solution χ of the underlying Simultaneous Chinese Remainder Problem.

Remark 15.4. *Remark 14.7 guarantees that the restricted analysis of D on the interval $\left(-\frac{M}{2} + m_{\max}, \frac{M}{2}\right]$ does not counterfeit the upcoming results. Theorem 12.2 indicates that the smallest solution can only be expected to lie in this interval if all remainder sets are singletons. Furthermore, elements from this set produce relatively large function outputs in absolute value such that it is unlikely that a lattice minimum with such an entry exists.*

We note that if

$$\chi \equiv \sum_{i=1}^{k} \sum_{j=1}^{t_i} \eta_{i,j} r_{i,j} b_i \mod M, \tag{15.18}$$

then,

$$D^{(\gamma)}(\chi) \equiv \sum_{i=1}^{k} \sum_{j=1}^{t_i} \eta_{i,j} r_{i,j} k_i^{(\gamma)} \mod SQ^{(\gamma)} \tag{15.19}$$

for all $\gamma \in \{1, \ldots, \Gamma\}$. This conclusion can be obtained by inserting Equation (15.18) into

$$D^{(\gamma)}(\chi) = \left[k_1^{(\gamma)} \chi_1 + \ldots, k_k^{(\gamma)} \chi_k \mod SQ^{(\gamma)} \right] \tag{15.20}$$

where $\chi_i = [\![\chi \mod m_i]\!]$ and observing that $\chi_i \equiv \sum_{j=1}^{t_i} \eta_{i,j} r_{i,j}$ as $b_i \equiv \delta_{i,j} \mod m_j$. We define our lattice by extending the Bleichenbacher-Nguyen lattice with the additional functional moduli and their functional coefficients. This leads to the $(T_\Sigma + 1 + \Gamma) \times (T_\Sigma + 1 + \Gamma)$ row lattice

$$\Lambda = \mathcal{L}\begin{pmatrix} M & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ r_{1,1}b_1 & B & 0 & \dots & 0 & r_{1,1}k_1^{(1)} & r_{1,1}k_1^{(2)} & \dots & r_{1,1}k_1^{(\Gamma)} \\ r_{1,2}b_1 & 0 & B & \dots & 0 & r_{1,2}k_1^{(1)} & r_{1,2}k_1^{(2)} & \dots & r_{1,1}k_1^{(\Gamma)} \\ \vdots & \vdots & & \ddots & \vdots & \vdots & \vdots & & \vdots \\ r_{k,t_k}b_k & 0 & 0 & \dots & B & r_{k,t_k}k_k^{(1)} & r_{k,t_k}k_k^{(2)} & \dots & r_{k,t_k}k_k^{(\Gamma)} \\ 0 & 0 & 0 & \dots & 0 & SQ^{(1)} & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & SQ^{(2)} & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & SQ^{(\Gamma)} \end{pmatrix} \tag{15.21}$$

whose elements are given by

$$(X, \eta_{1,1}B, \eta_{1,2}B, ..., \eta_{k,t_k}B, X^{(1)}, \dots, X^{(\Gamma)}) \in \mathbb{Z}^{T_\Sigma+1+\Gamma} \tag{15.22}$$

such that

$$X \equiv \sum_{i=1}^{k}\sum_{j=1}^{m} \eta_{i,j} r_{i,j} b_i \mod M \tag{15.23}$$

and

$$X^{(\gamma)} \equiv \sum_{i=1}^{k}\sum_{j=1}^{m} \eta_{i,j} r_{i,j} k_i^{(\gamma)} \mod SQ^{(\gamma)} \tag{15.24}$$

for all $\gamma \in \{1, \dots, \Gamma\}$. To limit the possibility of bad recombinations, we impose again that any combination satisfies

$$\sum_{j=1}^{t_{i_1}} \eta_{i_1,j} = \sum_{j=1}^{t_{i_2}} \eta_{i_2,j} \tag{15.25}$$

for all $1 \leq i_1, i_2 \leq k$ which is achieved by the following intersection

$$\widetilde{\Lambda} = \Lambda \cap \mathcal{L}\begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & B & 0 & \dots & 0 & B & 0 & \dots & 0 & \dots & B & 0 & \dots & 0 \\ 0 & 0 & B & \dots & 0 & B & 0 & \dots & 0 & \dots & B & 0 & \dots & 0 \\ \vdots & \vdots & & \ddots & & \vdots & \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & B & B & 0 & \dots & 0 & \dots & B & 0 & \dots & 0 \\ 0 & B & 0 & \dots & 0 & B & 0 & \dots & 0 & \dots & B & 0 & \dots & 0 \\ 0 & B & 0 & \dots & 0 & 0 & B & \dots & 0 & \dots & B & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & & \ddots & & & \vdots & \vdots & & \vdots \\ 0 & B & 0 & \dots & 0 & 0 & 0 & \dots & B & \dots & B & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots & & & \ddots & \vdots & \vdots & & \vdots \\ 0 & B & 0 & \dots & 0 & B & 0 & \dots & 0 & \dots & B & 0 & \dots & 0 \\ 0 & B & 0 & \dots & 0 & B & 0 & \dots & 0 & \dots & 0 & B & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & & \ddots & \\ 0 & B & 0 & \dots & 0 & B & 0 & \dots & 0 & \dots & 0 & 0 & \dots & B \end{pmatrix}. \tag{15.26}$$

We note that the matrix in Equation (15.26) is only a generating set and not a basis of $\mathcal{L}$. Furthermore, we observe that $\widetilde{\Lambda}$ contains the target vector

$$\tau := \Big(\chi, \eta_{1,1}B, \eta_{1,2}B, ..., \eta_{k,t_k}B, D^{(1)}(\chi), \dots, D^{(\Gamma)}(\chi)\Big) \tag{15.27}$$

where for all $\gamma \in \{1, \dots, \Gamma\}$, $D^{(\gamma)}$ denotes the centrally symmetric function corresponding to the modulus $SQ^{(\gamma)}$ defined in Chapter 14. The target vector τ is comparably short. Indeed, setting $\max\{SQ^{(1)}, \dots, SQ^{(\Gamma)}\} = SQ^{\max}$ and using the rough estimate $D^{\max}(\chi) \leq \chi\frac{SQ^{\max}}{M}$ from Equation (14.37) yields

$$\|\tau\|_2 \leq \sqrt{kB^2 + \chi^2 + \Gamma\chi^2\left(\frac{SQ^{\max}}{M}\right)^2}. \tag{15.28}$$

The determinant of Λ is $\det(\Lambda) = M \cdot B^{T_\Sigma}\prod_{i=1}^{\Gamma} SQ^{(i)}$ and so the Gaussian heuristic predicts a short vector of length

$$\lambda_1(\Lambda) \simeq \sqrt{\frac{T_\Sigma + \Gamma + 1}{2\pi e}}\det(\Lambda)^{\frac{1}{T_\Sigma+\Gamma+1}} \tag{15.29}$$

$$= \sqrt{\frac{T_\Sigma + \Gamma + 1}{2\pi e}}\left(M \cdot \prod_{i=1}^{\Gamma} SQ^{(i)} \cdot B^{T_\Sigma}\right)^{\frac{1}{T_\Sigma+\Gamma+1}}. \tag{15.30}$$

To put this analysis into context, we restrict to a particular set of moduli with nice properties. Set $\Gamma = k+1$, define $SQ^{(k+1)} = M_1 + \dots + M_k$ and let $SQ^{(i)} = SQ^{(k+1)} + M_i$ for all $i \in \{1, \dots, k\}$. Assuming that the initial moduli m_i were ordered in increasing order $m_1 < \dots < m_k$ yields that $\max\{D^{(1)}(\chi), \dots, D^{(k+1)}(\chi)\} = D^{(1)}$. Since $D^{(1)}(\chi) \leq \chi\frac{SQ^{(1)}}{M}$ and $SQ^{(1)} \leq (k+1)M_1$, this implies that $D^{(1)}(\chi) \leq \chi\frac{(k+1)}{m_1}$. Thus,

$$\|\tau\|_2 \leq \sqrt{kB^2 + \chi^2 + \chi^2\frac{(k+1)^3}{m_1^2}} \tag{15.31}$$

$$\leq B\sqrt{k+1+\frac{(k+1)^3}{m_1^2}} \tag{15.32}$$

$$\leq B\sqrt{k+1}\frac{m_1+k+1}{m_1}. \tag{15.33}$$

The determinant of Λ is $\det(\Lambda) = M \cdot SQ \cdot B^{T_\Sigma} \cdot \prod_{i=1}^{k} SQ^{(i)}$ and so the Gaussian heuristic predicts a short vector of expected length

$$\lambda_1(\Lambda) \simeq \sqrt{\frac{T_\Sigma + k + 2}{2\pi e}}\det(\Lambda)^{\frac{1}{T_\Sigma+k+2}} \tag{15.34}$$

$$= \sqrt{\frac{T_\Sigma + k + 2}{2\pi e}}(M \cdot SQ \cdot \prod_{i=1}^{k} SQ^{(i)} \cdot B^{T_\Sigma})^{\frac{1}{T_\Sigma+k+2}} \tag{15.35}$$

$$\leq \sqrt{\frac{T_\Sigma + k + 2}{2\pi e}}(M \cdot kM_k \cdot \prod_{i=1}^{k} (k+1)M_k \cdot B^{T_\Sigma})^{\frac{1}{T_\Sigma+k+2}} \tag{15.36}$$

$$\leq \sqrt{\frac{T_\Sigma + k + 2}{2\pi e}} M^{\frac{k+2}{T_\Sigma+k+2}} \cdot B^{\frac{T_\Sigma}{T_\Sigma+k+2}} \left(\frac{k}{m_k}\right)^{\frac{k+1}{T_\Sigma+k+2}}. \tag{15.37}$$

Thus, if $\sqrt{k+1}\frac{m_1+k+1}{m_1}\left(\frac{m_k}{k}\right)^{\frac{k+1}{T_\Sigma+k+2}} \leq \sqrt{\frac{T_\Sigma+k+2}{2\pi e}}\left(\frac{M}{B}\right)^{\frac{k+2}{T_\Sigma+k+2}}$, then the target vector has a high chance of being a shortest vector.

Remark 15.5. *Once again, the Gaussian heuristic should be used with caution. Indeed, our method still suffers from bad combinations caused by sufficiently small linear combinations of remainders. In particular, for all* r_{i,j_1}, r_{i,j_2}*, there exists a vector of length* $B\sqrt{r_{i,j_1}^2 + r_{i,j_2}^2}$ *obtained by subtracting the corresponding rows in Equation* (15.21)*. Thus, if* r_{i,j_1}, r_{i,j_2}, B *are sufficiently small, this vector is a short vector.*

Chapter 16

Reformulations

In this chapter, we reformulate some Simultaneous Chinese Remainder Problem variants.

16.1 A sieving problem

In this section, we display the General Bounded Simultaneous Chinese Remainder Problem from Definition 10.10 as a particular sieving problem. For this development, let $m_1 = p_1, ..., m_k = p_k$ be distinct primes and let for all $i \in \{1, ..., k\}$ $\mathcal{R}_i = \{r_{i,1}, \ldots, r_{i,t_i}\} \subseteq \{0, 1, ..., p_i - 1\}$ be a non-empty set of possible remainders modulo p_i. Let $M = \prod_{i=1}^{k} p_i$ and $M_i = \frac{M}{p_i}$ for all $i \in \{1, \ldots, k\}$. We consider the Simultaneous Chinese Remainder Problem instance

$$\mathsf{SimCRP}((p_1, \mathcal{R}_1), ..., (p_k, \mathcal{R}_k)) \tag{16.1}$$

with solutions in $\{0, \ldots, (\prod_{i=1}^{k} p_i) - 1\}$. Our presentation follows [CM05, Chapter 6] and [HH11, Chapter 2]. We refer to [For20] for an alternative description of the upcoming sieving results.

16.1.1 Intuition

Informally, a mathematical *sieve* requires a base set $\mathcal{A}$, a prime set $\mathcal{P}$ where for each prime $p \in \mathcal{P}$ a specific set $\mathcal{A}_p \subseteq \mathcal{A}$ of undesired elements is given, and an upper sieving bound z. It then approximates the number of elements in $\mathcal{A} \setminus \bigcup_{p \in \mathcal{P}} \mathcal{A}_p$. We note that this last quantity is related to the primitive solution set of a Simultaneous Chinese Remainder Problem. Indeed, if

- $\mathcal{A} = \{0, \ldots, (\prod_{i=1}^{k} p_i) - 1\}$,

- $\mathcal{P} = \{p_1, \ldots, p_k\}$, and
- $\mathcal{A}_{p_i} = \{a \in \mathcal{A} \mid [\![a \mod p_i]\!] \notin \mathcal{R}_i\}$,

then $\mathcal{A} \setminus \bigcup_{p \in \mathcal{P}} \mathcal{A}_p$ is the primitive solution set of

$$\mathsf{SimCRP}((p_1, \mathcal{R}_1), ..., (p_k, \mathcal{R}_k)). \tag{16.2}$$

If $\mathcal{A} = \{0, \ldots, B-1\}$ for some $B \in \{0, \ldots, (\prod_{i=1}^{k} p_i) - 1\}$, then $\mathcal{A} \setminus \bigcup_{p \in \mathcal{P}} \mathcal{A}_p$ represents the set of primitive solutions smaller than B. Thus, if $|\mathcal{A} \setminus \bigcup_{p \in \mathcal{P}} \mathcal{A}_p| > 0$, then the corresponding General Bounded Simultaneous Chinese Remainder Problem instance has an affirmative solution.

16.1.2 Auxiliary notations

Hereinafter, $\mathcal{A} = \mathcal{A}_1 \subseteq \mathbb{N}$ denotes any non-empty subset of natural numbers. $\mathcal{P}$ denotes a set of prime numbers. The letter p is used to denote a generic prime, and $z \in \mathbb{R}^+$ denotes a fixed positive real number. For any prime $p \in \mathcal{P}$, we fix an associated set $\mathcal{A}_p \subseteq \mathcal{A}$. For any square-free natural number d generated by primes of $\mathcal{P}$ only, let $\mathcal{A}_d = \bigcap_{p|d} \mathcal{A}_p$ be the intersection of the associated sets of the prime divisors of d. Let $P(z) = \prod_{\substack{p \in \mathcal{P} \\ p<z}} p$ be the product of all primes from $\mathcal{P}$ that are smaller than z. Let

$$S(\mathcal{A}, \mathcal{P}, z) = \left| \mathcal{A} \setminus \bigcup_{p|P(z)} \mathcal{A}_p \right| \tag{16.3}$$

be the number of elements in $\mathcal{A}$ that do not belong to any associated set $\mathcal{A}_p$ for all p dividing $P(z)$.

Remark 16.1. *Choosing the sets $\mathcal{A}, \mathcal{P}, \mathcal{A}_{p_i}$ for all $i \in \{1, \ldots, k\}$ as indicated at the end of Section 16.1.1 and setting $z > \max_{i \in \{1,\ldots,k\}} p_i$, shows that $S(\mathcal{A}, \mathcal{P}, z)$ is the number of primitive solutions of Equation (16.2) smaller than B.*

Additionally, let $\mu : \mathbb{N}_{\geq 1} \to \{-1, 0, 1\}$ denote the multiplicative Möbius function defined by $\mu(1) = 1$, $\mu(p) = -1$, $\mu(p^\alpha) = 0$ for all $\alpha \geq 2$ so that $\mu(p_1 ... p_k) = (-1)^k$. Let $\nu : \mathbb{N}_{\geq 1} \to \mathbb{N}_{\geq 1}$ be the prime divisor function counting the number of distinct prime divisors of a given natural number.

16.1.3 Elementary sieve results

Using the notations from Section 16.1.2, the following fundamental result on sieves can be deduced.

Theorem 16.2 (Theorem 6.2.1 in [CM05]). *For any function g such that $g(1) = 1$, we have*

$$S(\mathcal{A}, \mathcal{P}, z) = \sum_{d|P(z)} \mu(d)g(d)|\mathcal{A}_d| - \sum_{d|P(z)} \sum_{\substack{p|P(z)\\ p<q(d)}} \mu(d)(g(d) - g(pd))S(\mathcal{A}_{pd}, \mathcal{P}, p)$$

where $q(d)$ denotes the smallest prime divisor of d with the convention that $q(1) = +\infty$.

Theorem 16.2 is remarkable as it generalizes a variety of classical results. Indeed, setting $g(d) = 1$ for all $d \in \mathbb{N}$ leads to the *inclusion-exclusion principle*

$$S(\mathcal{A}, \mathcal{P}, z) = \sum_{d|P(z)} \mu(d)|\mathcal{A}_d|, \tag{16.4}$$

and setting $g(1) = 1$ and $g(d) = 0$ for all $d > 1$ leads to *Buchstab's identity*

$$S(\mathcal{A}, \mathcal{P}, z) = |\mathcal{A}| - \sum_{p|P(z)} S(\mathcal{A}_p, \mathcal{P}, p). \tag{16.5}$$

To illustrate these results, consider the following Simultaneous Chinese Remainder Problem

$$\mathsf{SimCRP}((3, \{1, 2\}), (5, \{2, 3\}), (7, \{3, 4, 5\})). \tag{16.6}$$

Setting

- $\mathcal{A} = \{0, \ldots, 104\}$,
- $\bar{\mathcal{A}}_3 = \{0\}$,
- $\bar{\mathcal{A}}_5 = \{0, 1, 4\}$, and
- $\bar{\mathcal{A}}_7 = \{0, 1, 2, 6\}$,

allows to define $\mathcal{A}_p = \{x \in A \mid [\![x \mod p]\!] \in \bar{\mathcal{A}}_p\}$ for all $p \in \{3, 5, 7\}$. Setting $\mathcal{A}_d = \cap_{p|d}\mathcal{A}_p$ and observing that $|\mathcal{A}_d| = \frac{|\mathcal{A}|}{d}|\bar{\mathcal{A}}_d|$, we can compute $S(\mathcal{A}, \{3, 5, 7\}, z)$ which corresponds to the number of primitive solutions of

Equation (16.6). Indeed, the inclusion-exclusion principle correctly predicts that for $z > 7$ there are

$$\begin{aligned}
&S(\mathcal{A}, \{3,5,7\}, z) && (16.7)\\
&= \sum_{d|105} \mu(d)|\mathcal{A}_d| && (16.8)\\
&= |\mathcal{A}| - |\mathcal{A}_3| - |\mathcal{A}_5| - |\mathcal{A}_7| + |\mathcal{A}_{15}| + |\mathcal{A}_{21}| + |\mathcal{A}_{35}| - |\mathcal{A}_{105}| && (16.9)\\
&= 105 - 35 - 63 - 60 + 21 + 20 + 36 - 12 && (16.10)\\
&= 12 && (16.11)
\end{aligned}$$

solutions to Equation (16.6) inside $\mathcal{A}$. Buchstab's identity yields the same conclusion, but interchanges the summation order:

$$\begin{aligned}
&S(\mathcal{A}, \{3,5,7\}, z) && (16.12)\\
&= |\mathcal{A}| - \sum_{p|P(z)} S(\mathcal{A}_p, \mathcal{P}, p) && (16.13)\\
&= |\mathcal{A}| - S(\mathcal{A}_3, \{3,5,7\}, 3) - S(\mathcal{A}_5, \{3,5,7\}, 5) - S(\mathcal{A}_7, \{3,5,7\}, 7) && (16.14)\\
&= |\mathcal{A}| - (|\mathcal{A}_3|) - (|\mathcal{A}_5| - |\mathcal{A}_{15}|) - (|\mathcal{A}_7| - |\mathcal{A}_{21}| - |\mathcal{A}_{35}| + |\mathcal{A}_{105}|) && (16.15)\\
&= 12. && (16.16)
\end{aligned}$$

We note that these computations used exact information on the sets $\mathcal{A}_d$ which are usually not available. Indeed, the elegant conclusion $|\mathcal{A}_d| = \frac{|\mathcal{A}|}{d}|\bar{\mathcal{A}}_d|$ stems from our particular choice of $\mathcal{A}$. However, if $\mathcal{A}$ is chosen differently, this exact information can only be determined by computing $\mathcal{A}_d$ explicitly which would trivially solve the underlying Simultaneous Chinese Remainder Problem. Thus, usually, $|\mathcal{A}_d|$ is only determined approximately. For example, $\left\lfloor \frac{|\mathcal{A}|}{d} \right\rfloor |\bar{\mathcal{A}}_d| \leq |\mathcal{A}_d| \leq \left\lceil \frac{|\mathcal{A}|}{d} \right\rceil |\bar{\mathcal{A}}_d|$ where $\bar{\mathcal{A}}_d$ denotes the undesired remainders modulo d. As each combination of primes needs to be considered, an exponential number of errors needs to be handled.

The computations can be simplified by considering only the most important terms of the sum in Equation (16.4). For example, setting $g(d) = 1$ whenever the number of distinct prime divisors of d is lower than 1 and $g(d) = 0$ otherwise (i.e., $\nu(d) > 1$) leads to the equality

$$S(\mathcal{A}, \mathcal{P}, z) = |\mathcal{A}| - \sum_{p|P(z)} |\mathcal{A}_p| + \sum_{p|P(z)} \sum_{\substack{p_1|P(z)\\ p_1<p}} S(\mathcal{A}_{pp_1}, \mathcal{P}, p_1). \qquad (16.17)$$

Recursively applying the same strategy to the rightmost sum reveals gradually more terms of the inclusion-exclusion principle. As the rightmost sum is always non-negative, this approach yields increasingly precise lower bounds for $S(\mathcal{A},\mathcal{P},z)$. These bounds are called *lower Bonferroni bounds* for $S(\mathcal{A},\mathcal{P},z)$. For example, the first Bonferroni bound is obtain as

$$S(\mathcal{A},\mathcal{P},z) \geq |\mathcal{A}| - \sum_{p|P(z)} |\mathcal{A}_p|. \tag{16.18}$$

the second one by

$$S(\mathcal{A},\mathcal{P},z) \geq |\mathcal{A}| - \sum_{p|P(z)} |\mathcal{A}_p| + \sum_{p|P(z)} \sum_{\substack{p_1|P(z)\\ p_1<p}} \Big(|\mathcal{A}_{pp_1}| - \sum_{p_2|P(p_1)} |A_{pp_1p_2}|\Big). \tag{16.19}$$

and the n-th Bonferroni bound by

$$S(\mathcal{A},\mathcal{P},z) \geq \sum_{\substack{d|P(z)\\ \nu(d)<2n-1}} \mu(d)|\mathcal{A}_d|. \tag{16.20}$$

Yet again, the difficulty lies in determining $|\mathcal{A}_d|$ for all the square-free divisors d considered inside the sums.

16.1.4 Brun's sieve

A good approximation of $S(\mathcal{A},\mathcal{P},z)$ using Theorem 16.2 requires an adequate error management. This may be achieved by a particular choice of the function g in Theorem 16.2 which simultaneously minimizes the number and the size of the errors. An extraordinarily creative choice of g allowed *Viggo Brun* [Bru15] to prove the following theorem.

Theorem 16.3 (Brun's sieve)**.** *Let $\omega : \mathbb{N} \to \mathbb{N}$ be a multiplicative function and assume that for all square-free d composed of primes of $\mathcal{P}$ only, we have $|\mathcal{A}_d| = \frac{\omega(d)}{d}|\mathcal{A}| + R_d$ for some $R_d \in \mathbb{Q}$. Let $W(z) = \prod_{p|P(z)} \left(1 - \frac{\omega(p)}{p}\right)$, let $b \in \mathbb{N}_{\geq 1}$ and let $\lambda \in \mathbb{R}^+$ be such that $0 < \lambda e^{1+\lambda} < 1$. Suppose that*

1. *$|R_d| \leq \omega(d)$ for all squarefree d composed by primes of $\mathcal{P}$ only,*

2. *there exists $A_1 \geq 1$ such that $0 \leq \frac{\omega(p)}{p} \leq 1 - \frac{1}{A_1}$, and*

3. *there exist $\kappa > 0$ and $A_2 \geq 1$ such that for all $2 \leq w \leq z$*

$$\sum_{w \leq p < z} \frac{\omega(p)\log(p)}{p} \leq \kappa \log\left(\frac{z}{w}\right) + A_2.$$

Let $\alpha = \frac{2\lambda}{\kappa}\frac{1}{1+\epsilon}$ where $\epsilon = \frac{1}{1+\frac{1}{200e^{1/\kappa}}}$. Then, for all sufficiently large z we have

$$S(\mathcal{A}, \mathcal{P}, z) > |A|W(z)\left(1 - \frac{2\lambda^{2b}e^{2\lambda}}{1-\lambda^2 e^{2+2\lambda}} e^{(2b+2)\frac{c_1}{\lambda \log(z)}}\right) - E$$

where $c_1 = \frac{A_2}{2}\left(1 + A_1\left(\kappa + \frac{A_2}{\log(2)}\right)\right)$ and $E = z^{2b-1+\frac{2.01}{e^{2\lambda/\kappa-1}}}$.

Remark 16.4. *Theorem 16.3 is an explicit version of [HH11, Theorem 2.1] that can be directly deduced from their proof by taking care of the error terms. Similarly, it can be shown that*

$$\log(z) \geq \max\left(\log(2)e^{\frac{2c_1 e^{\alpha}}{\kappa \log(2)\epsilon}}, 33\max(A_2, \kappa)\right) \tag{16.21}$$

is sufficiently large to deduce the desired conclusion.

Setting $\omega(p_i) = |\bar{\mathcal{A}}_{p_i}|$ for all $i \in \{1, \ldots, k\}$ where $\bar{\mathcal{A}}_{p_i} = \{0, \ldots, p_i - 1\} \setminus \mathcal{R}_i$ and multiplicatively extending ω to squarefree composites $d | P(\max p_i)$ reveals again the connection to the General Bounded Simultaneous Chinese Remainder Problem. Despeit its promising potential to solve the General Bounded Simultaneous Chinese Remainder Problem, Brun's sieve is impractical for most instances. Indeed, the required sieving bound z is generally large, which increases the error E. Thus, to guarantee a positive lower bound, $|\mathcal{A}|$ and $W(z)$ need to be sufficiently large. As $W(z)$ is fixed by the given moduli and $\mathcal{A} \subset \{0, \ldots, (\prod_{i=1}^k p_i) - 1\}$ cannot be arbitrarily extended, one can usually only deduce a trivial negative lower bound for $\mathcal{S}(\mathcal{A}, \mathcal{P}, z)$.

16.2 A subset sum problem

In this section, we formulate the Maximal Simultaneous Chinese Remainder Problem from Definition 11.3 as a particular subset sum problem. For this development, let $m_1, ..., m_k$ be pairwise coprime moduli and let for all $i \in \{1, ..., k\}$ $\mathcal{R}_i = \{r_{i,1}, \ldots, r_{i,t_i}\} \subseteq \{0, 1, ..., m_i - 1\}$ be a non-empty set of possible remainders modulo m_i. Let $M = \prod_{i=1}^k m_i$ and $M_i = \frac{M}{m_i}$ for all $i \in \{1, \ldots, k\}$. We consider the Simultaneous Chinese Remainder Problem instance

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), ..., (m_k, \mathcal{R}_k)) \tag{16.22}$$

with solutions in $\{0, \ldots, (\prod_{i=1}^k m_i) - 1\}$. Our presentation follows [KPP04].

16.2.1 The multiple-choice knapsack problem and its subset sum equivalent

The traditional *knapsack problem* puts forth a set of items $\mathcal{R}$, where each item has a weight w and a price p and asks to choose a subset of items such that the sum of prices is maximized subject to a knapsack capacity constraint claiming that the sum of weights cannot pass a fixed bound c. The multiple-choice knapsack problem generalizes this problem by considering multiple sets of items such that from each set exactly one item needs to be chosen.

Definition 16.5 (Multiple-choice knapsack problem)**.** Let $\mathcal{N}_1, \ldots, \mathcal{N}_k \subseteq \mathbb{R}^2$ be non-empty sets of items and let $c \in \mathbb{R}$ be a knapsack capacity. For all $i \in \{1, \ldots, k\}$ and each $j \in \{1, \ldots, |\mathcal{N}_i|\}$, let the item $\mathbf{v}_{i,j} = (p_{i,j}, w_{i,j}) \in \mathcal{N}_i$ have a profit $p_{i,j}$ and a weight $w_{i,j}$. Maximize $\sum_{i=1}^{k} \sum_{j=1}^{|\mathcal{N}_i|} \eta_{i,j} p_{i,j}$ subject to the constraints that $\sum_{i=1}^{k} \sum_{j=1}^{|\mathcal{N}_i|} \eta_{i,j} w_{i,j} \leq c$, $\sum_{j=1}^{|\mathcal{N}_i|} \eta_{i,j} = 1$ for all $i \in \{1, \ldots, k\}$, and $\eta_{i,j} \in \{0, 1\}$ for all $i \in \{1, \ldots, k\}$.

Usually, the prices $p_{i,j}$, the weights $w_{i,j}$, and the knapsack capacity c are non-negative integers, a convention that we adopt hereinafter. If $p_{i,j} = w_{i,j}$ for all $i \in \{1, \ldots, k\}$ and all $j \in \{1, \ldots, |\mathcal{N}_i|\}$, then we talk about the *multiple-choice subset sum problem.*

16.2.2 Problem construction

We observe the striking resemblance of the constraints from Definition 16.5 with Equation (15.9) and we use this similarity to reformulate the considered Simultaneous Chinese Remainder Problem instance into a multiple-choice subset-sum problem. Indeed, for all $i \in \{1, \ldots, k\}$, we let $b_i := \left[M_i^{-1} \mod m_i\right] M_i$ and set

$$\mathcal{N}_i = \{[\![r b_i \mod M]\!] \mid r \in \mathcal{R}_i\}. \tag{16.23}$$

Naturally, any solution χ of the considered Simultaneous Chinese Remainder Problem instance in Equation (16.22) satisfies

$$\chi \equiv \sum_{i=1}^{k} \sum_{j=1}^{t_i} \eta_{i,j} w_i \mod M \tag{16.24}$$

for some $w_i \in \mathcal{N}_i$ for all $i \in \{1, \ldots, k\}$ and the corresponding binary coefficients $\eta_{i,j}$. It remains to construct an integer problem. To do so, we introduce another value set

$$\mathcal{N}_* = \{0, M, 2M, \ldots, (k-1)M\} \tag{16.25}$$

and set the knapsack capacity to $c = kM - 1$. The sets $\mathcal{N}_1, \dots, \mathcal{N}_k, \mathcal{N}_*$ and c define our multiple-choice subset sum problem. By construction, if χ_1 the maximal solution to the Simultaneous Chinese Remainder Problem instance in Equation (16.22) and χ_2 is the solution of the considered subset sum problem, then

$$\chi_1 = [\![\chi_2 \mod M]\!]. \tag{16.26}$$

Remark 16.6. *We note that the subset-sum constructed from the Simultaneous Chinese Remainder Problem offers more structure than random subset sum problems. For example, all elements from one set $\mathcal{N}_i$ are divisible by b_i for all $i \in \{1, \dots, k\}$.*

Remark 16.7. *We can also construct an equivalent subset sum problem using the function D^+ from Proposition 14.3. By choosing $SQ < M$, we have a trade-off between the precision of the solution and the efficiency of the solving methods.*

16.2.3 Remarkable results

[Pis03] showed that a multiple-choice subset sum problem with $\mathcal{N}_1, \dots, \mathcal{N}_k$ and knapsack capacity c can be solved with a dynamic programming algorithm in time and space $O(T_\Sigma + \frac{c}{\log(c)})$ where $T_\Sigma = \sum_{i=1}^k |\mathcal{N}_i|$. This matches the complexity of the ordinary subset sum problem. [HLLP16] developed an approximate binary search algorithm that returns for all $t \in \mathbb{N}$, in time $O(T_\Sigma(t + \log(k))$, a solution z such that $\frac{z^* - z}{z^*} \leq \frac{2+\frac{1}{2^t}}{3+\frac{1}{2^t}}$ where z^* denotes the optimal solution. [Law77] developed a fully polynomial time approximation scheme finding a solution z such that $\frac{z^*-z}{z^*} \leq \epsilon$ in time $O(T_\Sigma log(T_\Sigma) + \frac{kT_\Sigma}{\epsilon})$.

We note that the large size of the solution that we expect for the multiple-choice subset-sum constructed in Section 16.2.2 makes these solving methods impractical. Indeed, the exact solver from [Pis03] has a time complexity of $O(\frac{kM}{\log(kM)})$ that is much higher than other solving methods that we studied in previous chapters. Considering the approximate solvers, we observe that ϵ in $\frac{z^*-z}{z^*} \leq \epsilon$ shrinks with increasing k. Indeed, as $z^* < kM$ we deduce that $\frac{z^*-z}{kM} \leq \frac{z^*-z}{z^*}$. To obtain a non-trivial solution, we require that $z^* - z \leq M$. Thus, $\epsilon \leq \frac{1}{k}$. Thereby, [HLLP16] is not suitable for our purpose and [Law77] can only be used for sufficiently small ϵ, which increases the time complexity. For example, if $z^* - z \leq \epsilon\sqrt{M}$, then we require $\epsilon \leq \frac{1}{k\sqrt{M}}$, which implies a time complexity of $O(T_\Sigma log(T_\Sigma) + k^2 T_\Sigma \sqrt{M})$.

Chapter 17

An illustrative example

In this chapter, we summarize our previous development through a concrete Simultaneous Chinese Remainder Problem instance. Indeed, consider

$$\mathsf{SimCRP}((3,\{1,2\}),(5,\{2,3\}),(7,\{3,4,5\}),\mathcal{S}_M) \tag{17.1}$$

and let $M = 3 \cdot 5 \cdot 7 = 105$. As the moduli are coprime, solutions are guaranteed and the Existential Simultaneous Chinese Remainder Problem from Definition 10.1 is trivial. If $S_M = \mathbb{Z} \cap \left(-\frac{M}{2}, \frac{M}{2}\right] = \{-52, \ldots, 52\}$, then its primitive solution set is

$$\{-52,-38,-37,-32,-23,-17,-2,17,32,38,47,52\} \tag{17.2}$$

showing that the Bounded Simultaneous Chinese Remainder Problem in Definition 10.7 has an affirmative solution for any bound $B > 2$. If $S_M = \mathbb{Z} \cap [0, M) = \{0, \ldots, 104\}$, then its primitive solution set is

$$\{17,32,38,47,52,53,67,68,73,82,88,103\} \tag{17.3}$$

showing that the General Bounded Simultaneous Chinese Remainder Problem in Definition 10.10 has an affirmative solution if and only if $B > 17$. Focusing on the latter problem setup, we note that the minimal solution is 17 and the maximal one 103. We note that

$$\frac{2-1}{5-1} < \frac{3-1}{7-1} < \frac{2-1}{3-1}. \tag{17.4}$$

such that under the optimal ordering from Proposition 12.4, Theorem 12.2 predicts that there are at least 12 solutions smaller than 105, 6 smaller than 70, 2 smaller than 60 and 1 smaller than 59. We observe that the given upper bounds are not tight which shows a general deficiency. To find the minimal solution, Chapter 13 suggests first computing the 4 solutions modulo 15 by combining the remainder information modulo 3 and 5 and subsequently computing the minimal solution through at most 10 comparisons. A direct search for the minimal solution would require the computation of 12 solutions modulo 105 and 11 comparisons. Using the development of Chapter 14, we observe that for

$$SQ = 5 \cdot 7 + 3 \cdot 7 + 3 \cdot 5 = 71, \tag{17.5}$$

the function

$$D^{+} : \{1, 2\} \times \{2, 3\} \times \{3, 4, 5\} \to \{0, \ldots, 70\} \tag{17.6}$$

defined by $D^{+}(x_3, x_5, x_7) = [\![47x_3 + 14x_5 + 10x_7 \mod SQ]\!]$ is minimized in $D^{+}(2, 2, 3) = 10$, which corresponds to the minimal solution 17. On the other hand, the function

$$D : \{1, 2\} \times \{2, 3\} \times \{3, 4, 5\} \to \{0, \ldots, 70\} \tag{17.7}$$

defined by

$$D(x_3, x_5, x_7) = [47x_3 + 14x_5 + 10x_7 \mod SQ] \tag{17.8}$$

is minimized in $D(1, 3, 5) = -3$, which corresponds to the maximal solution $103 \equiv -2 \mod 105$ and consists in the smallest solution in absolute value. Coppersmith's method from Section 15.1 constructs the polynomial

$$\mathcal{P}(x) := x^3 + 23x^2 + 26x + 73 \mod 105, \tag{17.9}$$

and, as $|-2| \leq 4 < 105^{\frac{1}{3}} = B$, it successfully retrieves the maximal solution. However, it is not guaranteed to find the minimal solution 17. Due to the small size of the problem, neither the Bleichenbacher-Nguyen method from Section 15.2.1 nor our improvement from Section 15.2.3 extend the bound B. However, for comparison, we note that our improvement yields a bound $B \simeq 0.165$ whereas Bleichenbacher-Nguyen yields $B \simeq 0.0197$. In practice, both methods find the maximal and minimal solution. The running example of Section 16.1 shows that sieve techniques might be used to find all the solutions of Equation (17.1), but its small size hinders a theoretical conclusion on the sieving bounds. To construct the multiple-choice subset sum problem from Section 16.2.2, we set $\mathcal{N}_1 = \{35, 70\}$, $\mathcal{N}_2 = \{42, 63\}$, $\mathcal{N}_3 = \{45, 60, 75\}$, $\mathcal{N}_* = \{0, 105, 210\}$, and $c = 315$. The desired solution is $313 = 70 + 63 + 75 + 105$ corresponding to the maximal solution 103.

Chapter 18

Experiments and heuristics

To better understand the Minimal Simultaneous Chinese Remainder Problem, we complete our theoretical development with some experimental results. These results allow us to analyze the behaviour of the minimal solution of Simultaneous Chinese Remainder Problem instances and to create some heuristics. Hereinafter, we let $m_1, \ldots, m_k \geq 3$ denote pairwise coprime moduli, $M = \prod_{i=1}^{k} m_i$, and $M_i = \frac{M}{m_i}$ for all $i \in \{1, \ldots, k\}$. We use the distribution notions from Chapter 2.

18.1 Cumulative distribution

Using the pairwise coprime moduli $m_1, \ldots, m_k$, we can build $2^{\sum_{i=1}^{k} m_i}$ distinct set products

$$\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \subseteq \{0, \ldots, m_1 - 1\} \times \cdots \times \{0, \ldots, m_k - 1\}. \tag{18.1}$$

Removing every combination containing an empty set, we deduce that we can build $\prod_{i=1}^{k}(2^{m_i} - 1)$ distinct Simultaneous Chinese Remainder Problem instances. As the moduli are pairwise coprime, we know that each problem instance has exactly $\prod_{i=1}^{k} |\mathcal{R}_i|$ solutions. Computing the occurrence of an integer as a minimal solution reveals a particular frequency distribution. In Figure 18.1 we represent the counts of our study as scatter plots. A point on position (x, y) indicates that x has been counted y times as the minimal solution among the $\prod_{i=1}^{k}(2^{m_i} - 1)$ distinct Simultaneous Chinese Remainder Problem instances. We note that the corresponding frequency distribution has the same properties as the given scatter plots, but downscales the y-axis to the probability space $[0, 1]$ by dividing the count values by $\prod_{i=1}^{k}(2^{m_i} - 1)$. For comparison purposes, we represent the count values only.

Remark 18.1. *The data sets for the statistics in this chapter were computed using Sagemath 9.0 running on Python 3.7.3. The scatter plots were generated using GraphPad Version 9.*

The remarkable shape of our data resembles a geometric distribution (the discrete version of an exponential distribution). The geometric distribution is a discrete probability distribution that models the number of failures before the first success. Let $\omega : \Omega \longrightarrow \mathbb{R}$ denote a real-valued random variable following the geometric distribution. Then, ω is supported on the set $\mathbb{N}$ and its probability mass function is defined by

$$\mathbb{P}(\omega = k) = (1-p)^k p \tag{18.2}$$

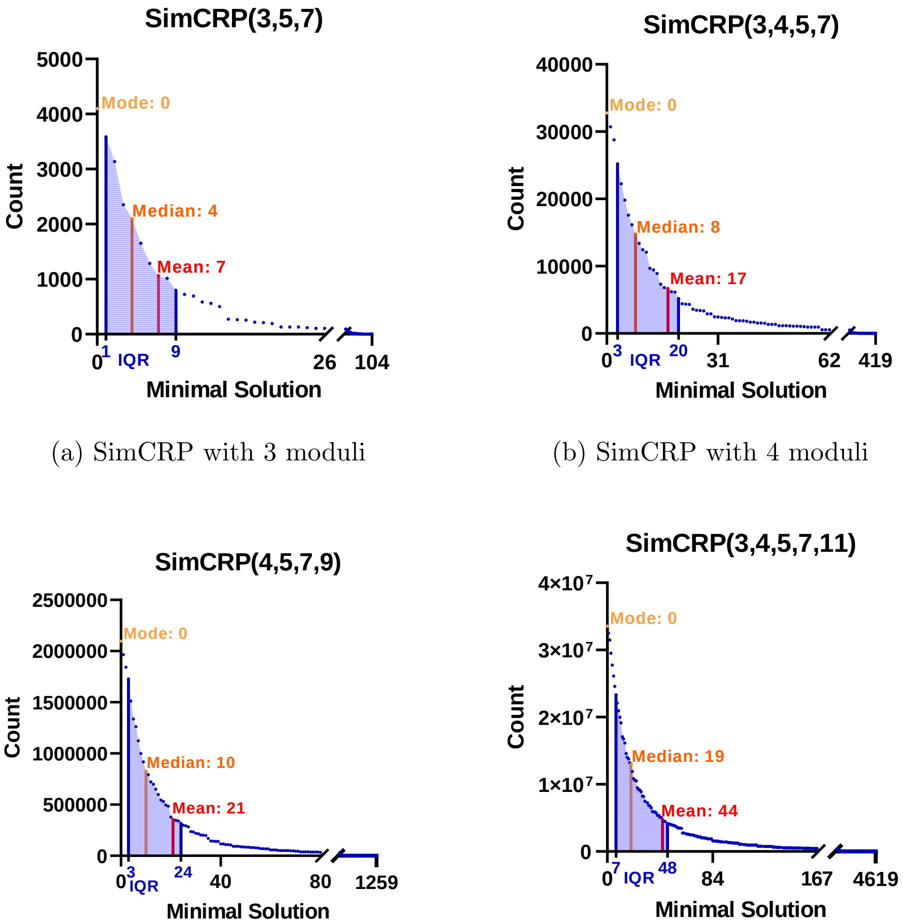
where p denotes the probability of success. The characteristics of the geometric distribution are that:

1. the *mode*, meaning the integer value with the highest probability, is 0,
2. the *mean* is $\mu = \frac{1-p}{p}$, and
3. the *median* is $\left\lceil \frac{-1}{\log_2(1-p)} \right\rceil - 1$.

To link the geometric distribution to our data, we note that Ω can be fixed as the set containing the $\prod_{i=1}^{k}(2^{m_i}-1)$ distinct Simultaneous Chinese Remainder Problem instances for a fixed set of pairwise coprime moduli $m_1, \ldots, m_k$. Assume that we employ a brute-force counting method to find the minimal solution of a Simultaneous Chinese Remainder Problem instance that starts at 0 and counts upwards until it finds a solution. Then, we can define ω such that it returns the number of iterations required by the brute-force counting method to solve a problem instance from Ω. Success means finding the minimal solution.

Remark 18.2. *We note that this description does not perfectly reflect the geometric distribution as the trials should take place under the same condition. However, in each iteration of the brute-force counting algorithm another integer is tested. Nonetheless, we will see that the geometric distribution fits well our data.*

Both, the geometric distribution and our data, are *right-skewed*, meaning that they have a longer *tail* of values on their right than on their left. In particular, the mode seems to be 0, the median is relatively low, and, due to the long tail, the average is shifted to the right of the median. To simplify our study, we focus in Section 18.2 on particular subsets of all Chinese Remainder Problem instances.



(a) SimCRP with 3 moduli

(b) SimCRP with 4 moduli

(c) SimCRP with 4 moduli

(d) SimCRP with 5 moduli

Figure 18.1: Scatter plots for the minimal solution of Simultaneous Chinese Remainder Problem instances with fixed moduli. An axis cut has been made at 95% of the counts. The mean is rounded to the closest integer and the rightmost entry on the horizontal axis represents the maximal minimal solution.

18.2 Fixed remainder set size

Let us restrict to Simultaneous Chinese Remainder Problem instances with remainder sets of a fixed size. Concretely, for all $i \in \{1, \ldots, k\}$, we consider $|\mathcal{R}_i| = t_i$ for some fixed $t_i \in \{1, \ldots, m_i\}$.

18.2.1 Closed form formula

Let

$$R(k) := \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \;\middle|\; \begin{array}{l} \forall i \in \{1, \ldots, k\} \\ |\mathcal{R}_i| = t_i \;\wedge\; \mathcal{R}_i \subseteq \{0, \ldots, m_i - 1\} \end{array} \right\} \tag{18.3}$$

denote the set of all remainder set combinations that can be obtained for the given remainder set sizes. We note that this set is in bijection with the Simultaneous Chinese Remainder Problem instances that can be obtained under the same conditions. By counting the number of combinations, we deduce that

$$|R(k)| = \prod_{i=1}^{k} \binom{m_i}{t_i}. \tag{18.4}$$

For all $n \in \{0, \ldots, M - 1\}$, let

$$S(n) := \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{array}{l} n \text{ is the minimal solution of} \\ \mathsf{SimCRP}((m_1, \mathcal{R}_1), \ldots, (m_k, \mathcal{R}_k)) \end{array} \right\}. \tag{18.5}$$

Then:

- 0 is the minimal solution if and only if $0 \in \mathcal{R}_i$ for all $i \in \{1, \ldots, k\}$. Thus, 0 is the minimal solution in

  $$|S(0)| = \prod_{i=1}^{k} \binom{m_i - 1}{t_i - 1} \tag{18.6}$$

  instances, where $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ denotes a binomial coefficient.

- 1 is the minimal solution if and only if $1 \in \mathcal{R}_i$ for all $i \in \{1, \ldots, k\}$ and $0 \notin \mathcal{R}_j$ for at least one $j \in \{1, \ldots, k\}$. Thus, 1 is the minimal solution in

  $$|S(1)| = \prod_{i=1}^{k} \binom{m_i - 1}{t_i - 1} - \prod_{i=1}^{k} f(m_i - 2, t_i - 2) \tag{18.7}$$

  instances, where $f(n, k) = \binom{n}{k}$ if $n \geq k \geq 0$, and $f(n, k) = 0$, otherwise. In particular, $|S(0)| \geq |S(1)|$.

- 2 is the minimal solution if and only if $2 \in \mathcal{R}_i$ for all $i \in \{1, \ldots, k\}$, $0 \notin \mathcal{R}_j$ for at least one $j \in \{1, \ldots, k\}$, and $1 \notin \mathcal{R}_j$ for at least one $j \in \{1, \ldots, k\}$. Thus, 2 is the minimal solution in
$$|S(2)| = \prod_{i=1}^{k} \binom{m_i - 1}{t_i - 1} - 2 \prod_{i=1}^{k} f(m_i - 2, t_i - 2) + \prod_{i=1}^{k} f(m_i - 3, t_i - 3) \quad (18.8)$$
instances. In particular, $|S(1)| \geq |S(2)|$.

In general, $n \in \{1, \ldots, M - 1\}$ is the minimal solution if and only if
$$\begin{cases} (n_1, \ldots, n_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \text{ and} \\ (a_1, \ldots, a_k) \notin \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \text{ for all } a \in \{0, \ldots, n-1\} \end{cases}$$
where for all $b \in \{0, \ldots, M-1\}$ and all $i \in \{1, \ldots, k\}$ we abuse our notation and set
$$b_i := [\![b \mod m_i]\!]. \quad (18.9)$$
Thus,
$$S(n) = \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{array}{l} (n_1, \ldots, n_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \\ \wedge\; T(n) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k = \emptyset \end{array} \right\}, \quad (18.10)$$
where $T(0) = \emptyset$ and for all $n \in \{1, \ldots, M - 1\}$
$$T(n) = \{(a_1, \ldots, a_k) \mid a \in \{0, \ldots, n - 1\}\}. \quad (18.11)$$
As the Equations (18.6)-(18.8) illustrate, the formula for the number of instances $|S(n)|$ satisfying these conditions gets increasingly complicated and we are not aware of a universal description for all $n \in \{0, \ldots, M - 1\}$.

18.2.2 Recursive counting function

Starting from a different perspective, a recursive counting method can be obtained. Concretely, we target to get a recursion on the number of remainder sets. To do so, we first generalize the notation in Equation (18.3) and fix for each $j \in \{1, \ldots, k\}$ the set
$$R(j) := \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_j \;\middle|\; \begin{array}{l} \forall i \in \{1, \ldots, j\} \\ |\mathcal{R}_i| = t_i \;\wedge\; \mathcal{R}_i \subseteq \{0, \ldots, m_i - 1\} \end{array} \right\}. \quad (18.12)$$
Subsequently, we adapt the sets in Equation (18.10) and Equation (18.11) to obtain
$$S(n, j, T) := \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_j \in R(j) \;\middle|\; \begin{array}{l} (n_1, \ldots, n_j) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_j \\ \wedge\; T \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_j = \emptyset \end{array} \right\}, \quad (18.13)$$
such that $S(n) = S(n, k, T(n))$, and
$$T \subseteq \{0, \ldots, m_1 - 1\} \times \cdots \times \{0, \ldots, m_j - 1\}. \quad (18.14)$$

Remark 18.3. *For technical reasons that appear in Equation* (18.19)*, we need to consider T in Equation* (18.14) *to be any subset of the cartesian product* $\{0,\ldots,m_1-1\}\times\cdots\times\{0,\ldots,m_j-1\}$.

As the set $\{\mathcal{R}_1\times\cdots\times\mathcal{R}_j\in R(j)\mid (n_1,\ldots,n_j)\in\mathcal{R}_1\times\cdots\times\mathcal{R}_j\}$, having cardinality $\prod_{i=1}^{j}\binom{m_i-1}{t_i-1}$, is the disjoint union of $S(n,j,T)$ and

$$S^C(n,j,T):=\left\{\mathcal{R}_1\times\cdots\times\mathcal{R}_j\in R(j)\;\middle|\;\begin{array}{l}(n_1,\ldots,n_j)\in\mathcal{R}_1\times\cdots\times\mathcal{R}_j\\ \wedge\, T\cap\mathcal{R}_1\times\cdots\times\mathcal{R}_j\neq\emptyset\end{array}\right\}, \tag{18.15}$$

we deduce that

$$\prod_{i=1}^{j}\binom{m_i-1}{t_i-1}=|S(n,j,T)|+|S^C(n,j,T)|. \tag{18.16}$$

Thus, to compute $|S(n,j,T)|$, it is sufficient to compute $|S^C(n,j,T)|$. We observe that for fixed $j>1$ and all $\mathcal{R}_1\times\cdots\times\mathcal{R}_j\in R(j)$, the intersection

$$T\cap\mathcal{R}_1\times\cdots\times\mathcal{R}_j\neq\emptyset \tag{18.17}$$

if and only if there exists $x_j\in\mathcal{R}_j$ such that

$$T_{x_j}\cap\mathcal{R}_1\times\cdots\times\mathcal{R}_{j-1}\neq\emptyset \tag{18.18}$$

where

$$T_{x_j}:=\left\{(x_1,\ldots,x_{j-1})\;\middle|\;\begin{array}{l}(x_1,\ldots,x_{j-1},x_j)\in T\,\wedge\\ \forall i\in\{1,\ldots,j-1\}\\ x_i\in\{0,\ldots,m_i-1\}\end{array}\right\}. \tag{18.19}$$

Put differently, Equation (18.17) is equivalent to

$$\bigcup_{x_j\in\mathcal{R}_j}T_{x_j}\cap\mathcal{R}_1\times\cdots\times\mathcal{R}_{j-1}\neq\emptyset. \tag{18.20}$$

This leads to the recursive counting formula

$$\left|S^C(n,j,T)\right|=\sum_{\substack{\mathcal{R}_j\subseteq\{0,\ldots,m_j-1\}\\ |\mathcal{R}_j|=t_j\\ n_j\in\mathcal{R}_j}}\left|S^C\left(n,j-1,\bigcup_{x_j\in\mathcal{R}_j}T_{x_j}\right)\right|. \tag{18.21}$$

To complete the recursion, we need to assure that the basis case can also be computed. If $j=1$, then Equation (18.16) yields that

$$\left|S^C(n,1,T)\right|=\binom{m_1-1}{t_1-1}-|S(n,1,T)| \tag{18.22}$$

and Equation (18.13) becomes

$$S(n, 1, T) = \{\mathcal{R}_1 \in R(1) \mid (n_1 \in \mathcal{R}_1) \ \wedge \ (T \cap \mathcal{R}_1 = \emptyset)\} \tag{18.23}$$

where $T \subseteq \{0, \ldots, m_1 - 1\}$. Clearly, if $n_1 \in T$, then $|S(n, 1, T)| = 0$. If $n_1 \notin T$ and $|T| > m_1 - t_1$, then any remainder set $\mathcal{R}_1 \in R(1)$ has a non-empty intersection with T. Indeed, otherwise $|T \cup \mathcal{R}_1| = |T| + |\mathcal{R}_1| > (m_1 - t_1) + t_1 = m_1$ which is impossible as $T \cup \mathcal{R}_1 \subseteq \{0, \ldots, m_1 - 1\}$. Thus, in this case, $|S(n, 1, T)| = 0$. If $n_1 \notin T$ and $|T| \leq m_1 - t_1$, then $\mathcal{R}_i$ contains n_1 and any combination of $t_1 - 1$ other elements that are not included in T. Thus, in this case, $|S(n, 1, T)| = \binom{m_1 - 1 - |T|}{t_1 - 1}$. In summary,

$$|S(n, 1, T)| = \begin{cases} 0 & \text{if } (n_1 \in T) \ \vee \ (|T| > m_1 - t_1), \\ \binom{m_1 - 1 - |T|}{t_1 - 1} & \text{otherwise}, \end{cases} \tag{18.24}$$

which completes the recursion.

18.2.3 A constructive approach

Although the recursive method from Section 18.2.2 computes $|S(n)|$ for each $n \in \{0, \ldots, n - 1\}$, it is not optimal. To be precise, with the current definition, the procedure needs to start anew for each n as no information from previous iterations is recycled. Furthermore, the complexity of the computations increases for increasing n as the initial set $T(n)$, defined by $T(0) = \emptyset$ and $T(n) = \{(a_1, \ldots, a_k) \mid a \in \{0, \ldots, n - 1\}\}$ for all $n > 0$, increases such that the resulting union $\bigcup_{x_j \in \mathcal{R}_j} T_{x_j}$ grows as well. Thereby, the number of elementary operations (comparisons and insertions) increases which makes the procedure less efficient for large n.

In this subsection, we prove that there is an inherent relation between $S^C(n) := S^C(n, k, T(n))$ and $S^C(n + 1)$ that will be exploited in Section 18.2.4 to design an improved counting method. Concretely, we prove that for all $n \in \{0, \ldots, M - 2\}$,

$$S^C(n + 1) = S^C_{+1}(n) \cup S^C(n + 1, k, T(1)) \tag{18.25}$$

where

$$S^C_{+1}(n) := \left\{\mathcal{R}_1^{+1} \times \cdots \times \mathcal{R}_k^{+1} \mid \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C(n)\right\} \tag{18.26}$$

with

$$\mathcal{R}_i^{+1} := \{[\![r_i + 1 \mod m_i]\!] \mid r_i \in \mathcal{R}_i\} \tag{18.27}$$

for all $i \in \{1, \ldots, k\}$.

First, we prove that $S^C_{+1}(n) \subseteq S^C(n+1)$. To do so, let

$$\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C(n), \tag{18.28}$$

where, by Equation (18.13),

$$S^C(n) = \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{matrix} (n_1, \ldots, n_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \\ \wedge\, T(n) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset \end{matrix} \right\}. \tag{18.29}$$

As $|\mathcal{R}_i| = t_i$, it also holds that $|\mathcal{R}_i^{+1}| = t_i$, such that

$$\mathcal{R}_1^{+1} \times \cdots \times \mathcal{R}_k^{+1} \in R(k) \tag{18.30}$$

and so $S^C_{+1}(n) \subseteq R(k)$. By Equation (18.27) and since

$$(n_1, \ldots, n_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k, \tag{18.31}$$

we also have

$$((n+1)_1, \ldots, (n+1)_k) \in \mathcal{R}_1^{+1} \times \cdots \times \mathcal{R}_k^{+1}. \tag{18.32}$$

Furthermore, as

$$T(n) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset, \tag{18.33}$$

there exists $a \in \{0, \ldots, n-1\}$ such that

$$(a_1, \ldots, a_n) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k. \tag{18.34}$$

By Equation (18.27), this implies that

$$((a+1)_1, \ldots, (a+1)_k) \in \mathcal{R}_1^{+1} \times \cdots \times \mathcal{R}_k^{+1}. \tag{18.35}$$

As $(a+1) \in \{1, \ldots, n\}$, we deduce that $((a+1)_1, \ldots, (a+1)_k) \in T(n+1)$, whereby

$$T(n+1) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset. \tag{18.36}$$

Thus,

$$\mathcal{R}_1^{+1} \times \cdots \times \mathcal{R}_k^{+1} \in S^C(n+1). \tag{18.37}$$

Since this argument holds for every $\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C(n)$, we deduce that

$$S^C_{+1}(n) \subseteq S^C(n+1), \tag{18.38}$$

as desired.

Next, we prove that $S^C_{+1}(n)$ contains almost all elements of $S^C(n+1)$. More precisely, we show that for all

$$\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C(n+1) \tag{18.39}$$

for which there exists $a \in \{1, \ldots, n\}$ such that

$$(a_1, \ldots, a_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \tag{18.40}$$

we have

$$\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C_{+1}(n). \tag{18.41}$$

Indeed, let

$$\mathcal{R}_i^{-1} := \{\llbracket r_i - 1 \mod m_i \rrbracket \mid r_i \in \mathcal{R}_i\}. \tag{18.42}$$

Then, as $|\mathcal{R}_i| = t_i$, it also holds that $|\mathcal{R}_i^{-1}| = t_i$, such that

$$\mathcal{R}_1^{-1} \times \cdots \times \mathcal{R}_k^{-1} \in R(k). \tag{18.43}$$

Since $\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C(n+1)$, we know that

$$((n+1)_1, \ldots, (n+1)_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k, \tag{18.44}$$

so that, by Equation (18.42),

$$(n_1, \ldots, n_k) \in \mathcal{R}_1^{-1} \times \cdots \times \mathcal{R}_k^{-1}. \tag{18.45}$$

Furthermore, as

$$(a_1, \ldots, a_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k, \tag{18.46}$$

Equation (18.42) implies that

$$((a-1)_1, \ldots, (a-1)_k) \in \mathcal{R}_1^{-1} \times \cdots \times \mathcal{R}_k^{-1}. \tag{18.47}$$

Since $a \in \{1, \ldots, n\}$, we deduce that $(a-1) \in \{0, \ldots, n-1\}$, which implies that $((a-1)_1, \ldots, (a-1)_k) \in T(n)$ revealing that

$$T(n) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset. \tag{18.48}$$

Thereby,

$$\mathcal{R}_1^{-1} \times \cdots \times \mathcal{R}_k^{-1} \in S^C(n). \tag{18.49}$$

Since $(\mathcal{R}_i^{-1})^{+1} = \mathcal{R}_i$, we deduce that

$$\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C_{+1}(n). \tag{18.50}$$

The last observation yields that $S^C_{+1}(n)$ contains all remainder set products that have a nonempty intersection with $T(n+1) \setminus T(1)$ where $T(1) = \{(0,\dots,0)\}$, which shows that

$$S^C_{+1}(n) = \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{array}{l} ((n+1)_1,\dots,(n+1)_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \\ \wedge\, (T(n+1) \setminus T(1)) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset \end{array} \right\}.$$

Thus, only remainder set products which include the tuple $(0,\dots,0)$ may not belong to $S^C_{+1}(n)$. Hence, by Equation (18.29), we conclude that

$$S^C(n+1) \tag{18.51}$$

$$= \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{array}{l} ((n+1)_1,\dots,(n+1)_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \\ \wedge\, T(n+1) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset \end{array} \right\} \tag{18.52}$$

$$= S^C_{+1}(n) \cup \tag{18.53}$$

$$\left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{array}{l} ((n+1)_1,\dots,(n+1)_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \\ \wedge\, T(1) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset \end{array} \right\} \tag{18.54}$$

$$= S^C_{+1}(n) \;\cup\; S(n+1,k,T(1)) \tag{18.55}$$

where we used in Equation (18.55) the notation from Equation (18.15). This completes the proof of Equation (18.25).

Remark 18.4. *The union in Equation* (18.55) *may not be disjoint.*

18.2.4 Improved counting method

The constructive recursion in Equation (18.25) introduces an improved method to compute $|S(n+1)|$ for all $n \in \{0,\dots,M-2\}$ as essentially all the information from $S^C(n)$ can be recycled. However, using this recursion, $S^C(n)$ needs to be computed explicitly and updated in each iteration. Concretely, to compute $S^C(n+1)$, one needs to compute first $S^C_{+1}(n)$ from $S(n)$ through a simple modular shift of remainder set products. Then, $S^C(n+1,k,T(1))$ needs to be computed and merged with $S^C_{+1}(n)$. By Equation (18.15),

$$S^C(n+1,k,T(1)) \tag{18.56}$$

$$= \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{array}{l} (n_1,\dots,n_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \\ \wedge\, T(1) \cap \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \neq \emptyset \end{array} \right\} \tag{18.57}$$

$$= \left\{ \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in R(k) \;\middle|\; \begin{array}{l} (n_1,\dots,n_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \\ \wedge\, (0,\dots,0) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \end{array} \right\} \tag{18.58}$$

for all $n \in \{0,\dots,M-1\}$. We observe that $|S^C(n+1,k,T(1))|$ can be easily computed. Concretely, for all $i \in \{1,\dots,k\}$:

1. If $n_i = 0$, then there are $\binom{m_i-1}{t_i-1}$ choices for $\mathcal{R}_i$.

2. If $n_i \neq 0$ and $t_i = 1$, then there is no choice for $\mathcal{R}_i$, which implies that $S^C(n+1, j, T(1)) = \emptyset$.

3. If $n_i \neq 0$ and $t_i > 1$, there are $\binom{m_i-2}{t_i-2}$ choices for $\mathcal{R}_i$.

The choices in the above descriptions are obtained through the combinations of the elements in $\{0, \ldots, m_i - 1\} \setminus \{0, n_i\}$ which can be directly constructed.

We note that for increasing n, the set $S^C(n)$ increases as well. Thereby, the computation of $S^C_{+1}(n)$ becomes less efficient. On the contrary, the computation of $S(n+1, k, T(1))$ is almost independent from this increase. As, we expect that $|S^C_{+1}(n)| > |S(n+1, k, T(1))|$, for most $n \in \{0, \ldots, M-2\}$, our procedure could be optimized by limiting the number of operations needed to compute $S^C_{+1}(n)$, even if this increases the number of operations needed to compute $S^C(n+1, k, T(1))$. We note that we are only interested in the cardinality of the set $S^C(n)$ and so we may interchange it with a bijective equivalent whose computation is more efficient. To do so, we set for all $S \subseteq R(k)$ and all $n \in \{0, \ldots, M-1\}$ the notation

$$S_{-n} := \{\mathcal{R}_1^{-n} \times \cdots \times \mathcal{R}_k^{-n} \mid \mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S\}, \tag{18.59}$$

where for all $i \in \{1, \ldots, k\}$

$$\mathcal{R}_i^{-n} := \{[\![r_i - n \mod m_i]\!] \mid r_i \in \mathcal{R}_i\}. \tag{18.60}$$

Trivially, $|S| = |S_{-n}|$ which indicates that S is in bijection with S_{-n}. Furthermore, we remark that with this notation Equation (18.25) becomes

$$\begin{aligned} S^C_{-(n+1)}(n+1) &= (S^C_{+1})_{-(n+1)}(n) \cup S^C_{-(n+1)}(n+1, k, T(1)) && (18.61)\\ &= S^C_{-n}(n) \cup S^C_{-(n+1)}(n+1, k, T(1)). && (18.62) \end{aligned}$$

Thus, to compute $S^C_{-(n+1)}(n+1)$, the set $S^C_{-n}(n)$ is completely recycled without the need of changing its elements. Yet, it remains to compute the auxiliary set $S^C_{-(n+1)}(n+1, k, T(1))$ and merge it with $S^C_{-n}(n)$. We note that $S^C_{-(n+1)}(n+1, k, T(1))$ is obtained by first computing $S^C(n+1, k, T(1))$ as described in Equation (18.58) and then shifting all its remainder set products by $-(n+1)$. This leads to the simple pseudocode in Algorithm 18.1.

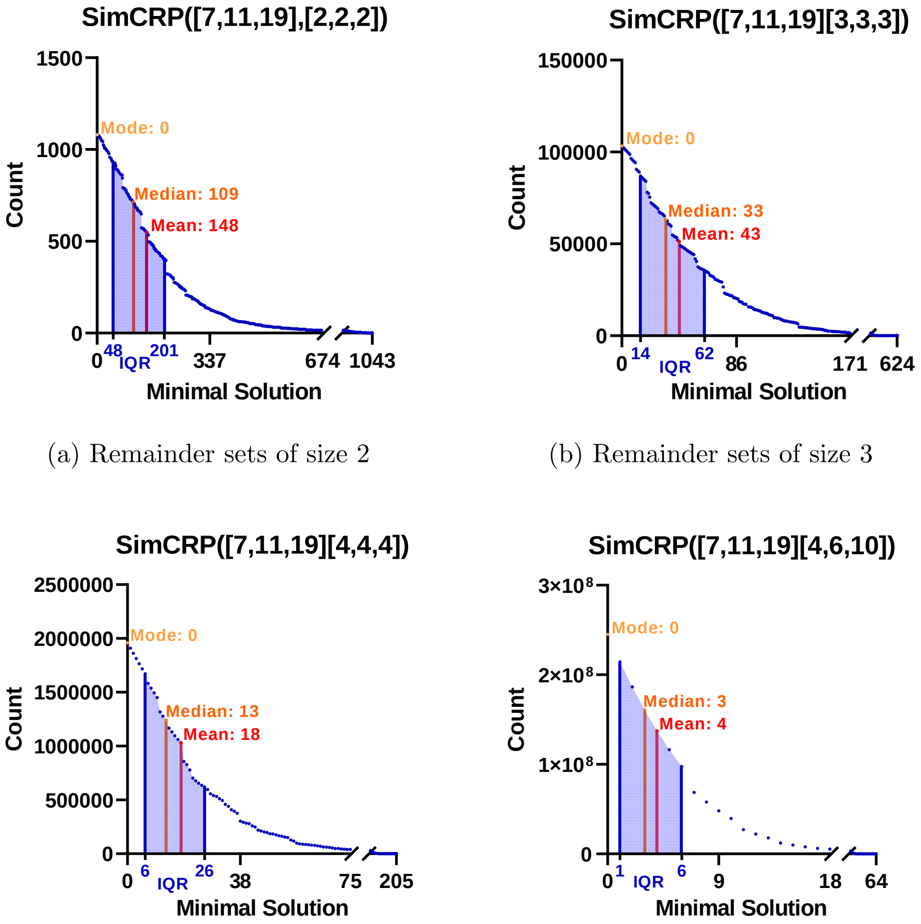
```
Algorithm 18.1: Minimal solution counter
   Input: Given a list of pairwise coprime moduli $(m_1, \ldots, m_k)$ and a
          list of remainder set sizes $(t_1, \ldots, t_k)$.
   Output: The algorithm computes a list indicating how often each
           $n \in \{0, \ldots, (\prod_{i=1}^{k} m_i) - 1\}$ is the minimal solution among
           all Simultaneous Chinese Remainder Problem instances
           with the given moduli and remainder set sizes.
1  Counts← $\left[\prod_{i=1}^{k} \binom{m_i-1}{t_i-1}\right]$;
2  $S^C = \emptyset$;
3  for $n = 0$ until $(\prod_{i=1}^{k} m_i) - 2$ do
4      for $\mathcal{R}_1 \times \cdots \times \mathcal{R}_k \in S^C(n+1, k, T(1))$ do
5          Shift← {};
6          for $(x_1, \ldots, x_k) \in \mathcal{R}_1 \times \cdots \times \mathcal{R}_k$ do
7              Shift← Shift $\cup\big\{([\![x_1 - (n+1) \mod m_1]\!], \ldots$
8                          $\ldots, [\![x_k - (n+1) \mod m_k]\!])\big\}$;
9          $S^C \leftarrow S^C \cup$ Shift;
10     Counts.append$\left((\prod_{i=1}^{k} \binom{m_i-1}{t_i-1}) - |S^C|\right)$
11 return Counts
```

18.2.5 Particular observations

By Equation (18.26), $S^C(n)$ is in bijection with $S^C_{+1}(n)$. Thus, Equation (18.25) implies that $|S^C(n+1)| \geq |S^C(n)|$ for all $n \in \{0, \ldots, M-2\}$. Hence, by Equation (18.16), the sequence $\{|S(n)|\}_{n \in \{0,\ldots,M-1\}}$ is decreasing for increasing n. In particular, this yields that 0 is the mode of the distribution with $|S(0)| = \prod_{i=1}^{k} \binom{m_i-1}{t_i-1}$ and that

$$|S(n)| = \prod_{i=1}^{k} \binom{m_i - 1}{t_i - 1} - |S^C(n)| \leq \prod_{i=1}^{k} \binom{m_i - 1}{t_i - 1} \tag{18.63}$$

for all $n \in \{0, \ldots, M-1\}$. Furthermore, this indicates that the "for" loop in line 3 of Algorithm 18.1 can be stopped if the first 0 is appended to the counting list in line 10 as no subsequent integer appears as the minimal solutions of a Simultaneous Chinese Remainder Problem instance with the given moduli and remainder set sizes.



(a) Remainder sets of size 2

(b) Remainder sets of size 3

(c) Remainder sets of size 4

(d) Remainder sets of varying sizes

Figure 18.2: Scatter plots for the minimal solution of Simultaneous Chinese Remainder Problem instances with fixed moduli and remainder sets of fixed sizes. An axis cut has been made at 99% of the counts. The mean is rounded to the closest integer and the rightmost entry on the horizontal axis represents the maximal minimal solution.

18.3 Simulating the distributions

In this section, we simulate our data with respect to the geometric distribution. To do so, we need to find a suitable probability parameter p that defines the simulating geometric distribution. As the cumulative distributions from Section 18.1 result as the sum of partial distributions as described in Section 18.2, we restrict our attention to the distribution of the minimal solution of Simultaneous Chinese Remainder Problem instances with fixed moduli $m_1, \ldots, m_k$ and fixed remainder set sizes $|\mathcal{R}_i| = t_i$ for all $i \in \{1, \ldots, k\}$. By Section 18.2.5 the mode of the corresponding distribution is 0, which holds also for the geometric distribution. Therefore, we simply define the desired probability parameter p such that the probability of obtaining 0 in the geometric distribution is the same as for the empirical distribution. To be precise, Equation (18.4) yields that there are $|R(k)| = \prod_{i=1}^{k} \binom{m_i}{t_i}$ Simultaneous Chinese Remainder Problem instances with the given remainder set sizes. Furthermore, by Equation (18.6), there are $|S(0)| = \prod_{i=1}^{k} \binom{m_i-1}{t_i-1}$ Simultaneous Chinese Remainder Problem instances with minimal solution 0. Thus, the probability of obtaining a Simultaneous Chinese Remainder Problem instance with minimal solution 0 by sampling the remainder sets uniformly at random from $R(k)$ is $\frac{|S(0)|}{|R(k)|} = \prod_{i=1}^{k} \frac{t_i}{m_i}$. As the probability of obtaining 0 in the geometric distribution is $\mathbb{P}(\omega = 0) = (1-p)^0 p = p$, we set

$$p := \prod_{i=1}^{k} \frac{t_i}{m_i}. \tag{18.64}$$

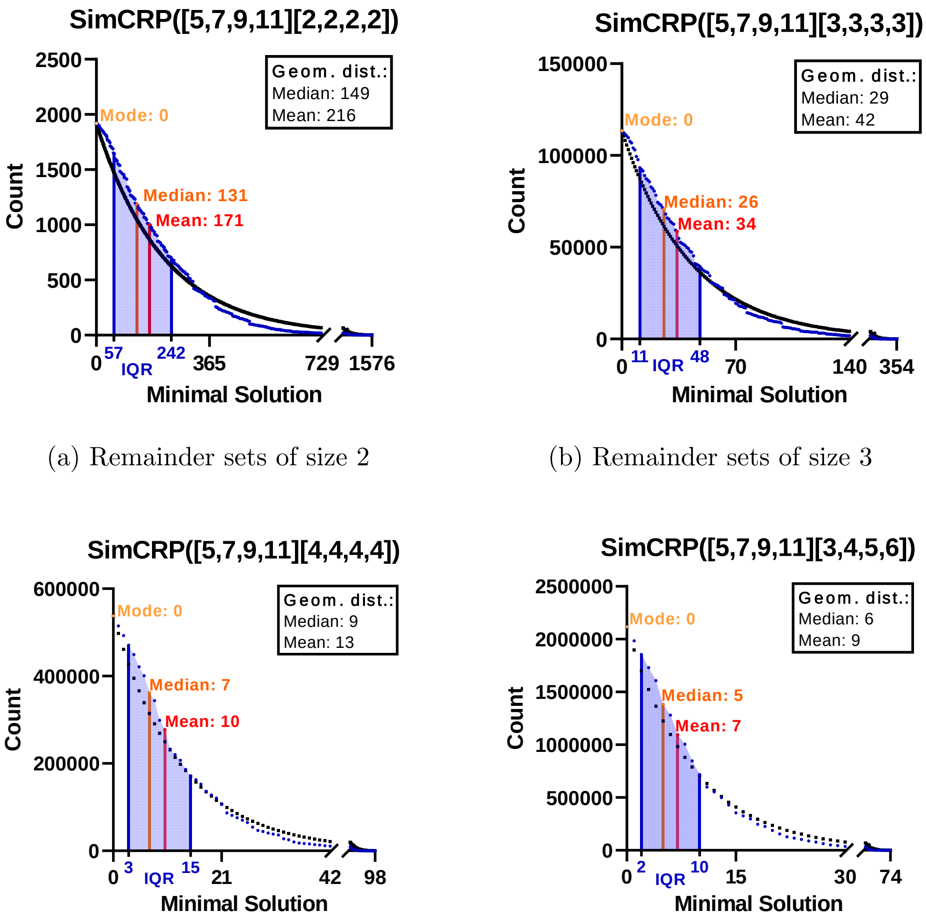
Thereby, the mean of the geometric distribution is

$$\mu := \frac{1-p}{p} = \frac{1}{p} - 1 = \left(\prod_{i=1}^{k} \frac{m_i}{t_i} \right) - 1 \tag{18.65}$$

and using the fact that $\log(1+x) \leq x$ for $x > -1$, the median is

$$q_{1/2} := \left\lceil \frac{-1}{\log_2(1-p)} \right\rceil - 1 \leq \frac{-\log(2)}{\log(1-p)} \leq \frac{\log(2)}{p} = \log(2) \prod_{i=1}^{k} \frac{m_i}{t_i}. \tag{18.66}$$

These formulas correspond to our intuition on the minimal solution of a Simultaneous Chinese Remainder Problem instance with fixed remainder set sizes. Indeed, assuming that the solutions are well-distributed in $\{0, \ldots, M-1\}$, we expect to find the minimal solution close to $\left(\prod_{i=1}^{k} \frac{m_i}{t_i} \right)$. Equation (18.65) confirms this intuition on average and Equation (18.66) yields that generally the minimal solution is even smaller.



(a) Remainder sets of size 2

(b) Remainder sets of size 3

(c) Remainder sets of size 4

(d) Remainder sets of varying sizes

Figure 18.3: Scatter plots for the minimal solution of Simultaneous Chinese Remainder Problem instances with fixed moduli and remainder sets of fixed sizes (blue) featuring a simulation using the geometric distribution (black). An axis cut has been made at 99% of the counts. The mean is rounded to the closest integer and the rightmost entry on the horizontal axis represents the maximal minimal solution.

18.4 The maximal minimal solution

Contrary to the geometric distribution, which assesses for each natural number $n \in \mathbb{N}$ a non-zero probability, our data sets have a maximal entry. This maximal entry, found as the rightmost value on the x-axis of our scatterplots, represents the maximal minimal solution of any Simultaneous Chinese Remainder Problem instance with remainder sets of the given sizes. We observe that the maximal minimal solution seems to be negatively proportional to the remainder set sizes. By inspection, we deduce that the maximal minimal solution for a Simultaneous Chinese Remainder Problem instance with remainder sets of size $|\mathcal{R}_i| = t_i$ for all $i \in \{1, \dots, k\}$ is often given by

$$\mathbf{M} := M - 1 - \sum_{i=1}^{k} (t_i - 1) M_i \tag{18.67}$$

with $M_i = \frac{M}{m_i}$. We note that if $\mathbf{M} \geq 0$, then $\mathbf{M}$ is the minimal solution of

$$\mathsf{SimCRP}((m_1, \mathcal{R}_1), \dots, (m_k, \mathcal{R}_k)) \tag{18.68}$$

where

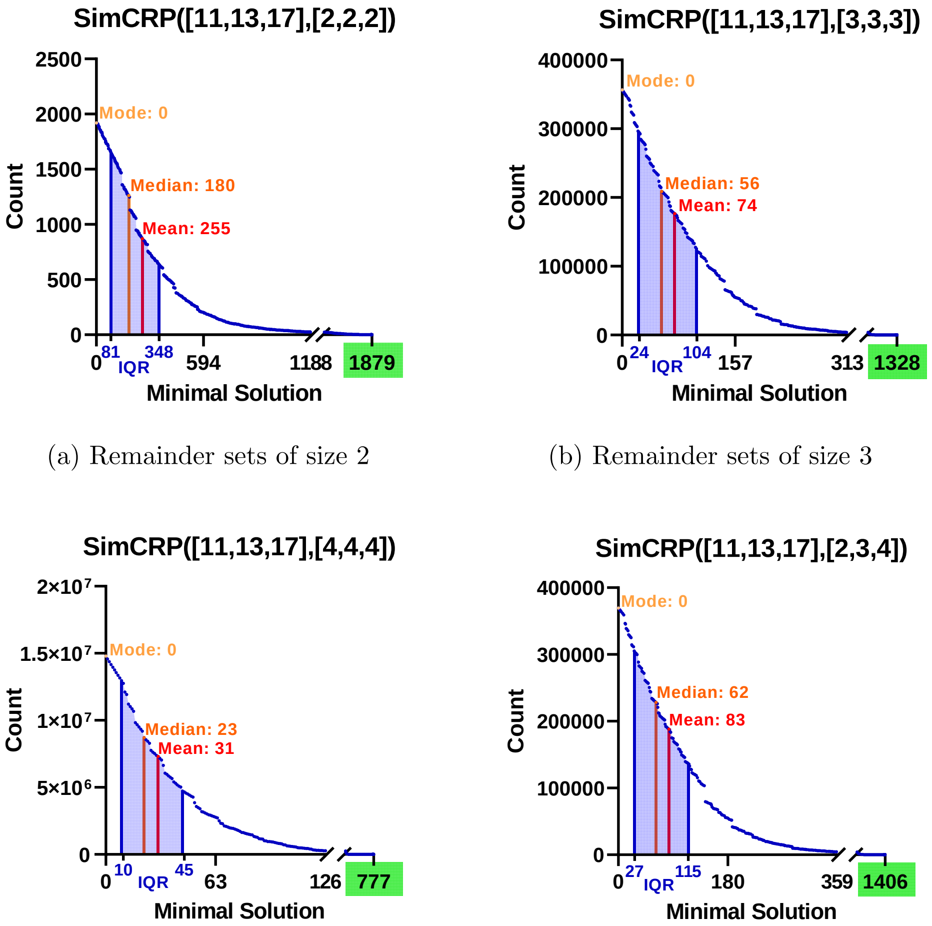
$$\mathcal{R}_i = \{[\![-1 - \psi_i M_i \mod m_i]\!] \mid \psi_i \in \{0, \dots, t_i - 1\}\} \tag{18.69}$$

for all $i \in \{1, \dots, k\}$. Indeed, this particular problem instance has the primitive solution set

$$\mathcal{S} := \left\{ M - 1 - \sum_{i=1}^{k} \psi_i M_i \mid \psi_i \in \{0, \dots, t_i - 1\} \; \forall i \in \{1, \dots, k\} \right\} \tag{18.70}$$

and so it contains $\mathbf{M}$. As any other integer in $\mathcal{S}$ is strictly larger than $\mathbf{M}$ and smaller than M, $\mathbf{M}$ is indeed its minimal solution. For example, Figure 18.4a shows that the maximal solution for the moduli $11, 13, 17$ and remainder sets of size 2 is given by $1879 = 2431 - 1 - 221 - 187 - 143$. This minimal solution is achieved by $\mathsf{SimCRP}((11, \{9, 10\}), (13, \{7, 12\}), (17, \{9, 16\}))$ having the primitive solution set $\{1879, 2022, 2066, 2100, 2209, 2243, 2287, 2430\}$.

Remark 18.5. *Our development only shows that* $\mathbf{M}$ *is a lower bound for the maximal minimal solution, but it does not rule out that a larger minimal solution exists. In particular, if* $\mathbf{M} < 0$*, the resulting lower bound is trivial. Yet,* $\mathbf{M}$ *seems to be the maximal minimal solution for a non-negligible portion of all remainder set sizes. This indicates that the proven upper bound from Theorem 12.2 may be several magnitudes too large.*



(a) Remainder sets of size 2

(b) Remainder sets of size 3

(c) Remainder sets of size 4

(d) Remainder sets of varying sizes

Figure 18.4: Scatter plots for the minimal solution of Simultaneous Chinese Remainder Problem instances with fixed moduli and remainder sets of fixed sizes. An axis cut has been made at 99% of the counts. The mean is rounded to the closest integer and the rightmost entry on the horizontal axis represents the maximal minimal solution.

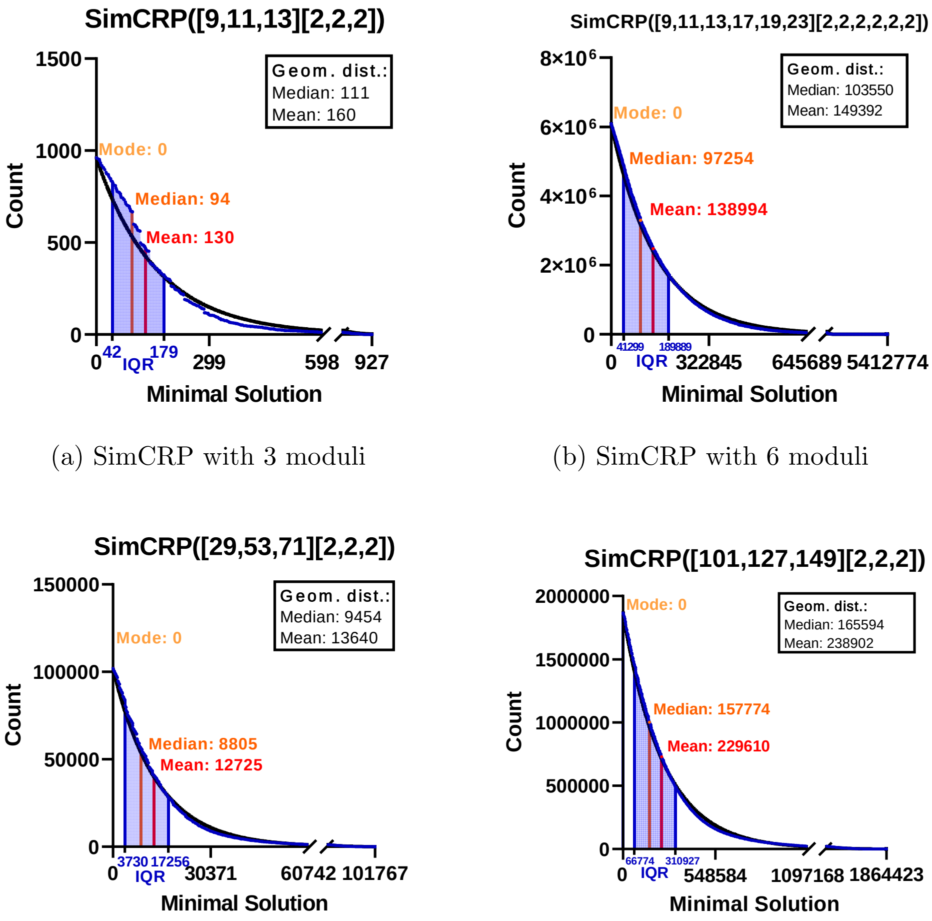
18.5 Remainder sets of size two

Figure 18.4 confirms our intuition from Remark 9.3 claiming that it seems to be easier to find the minimal solution of a Simultaneous Chinese Remainder Problem instance with large remainder set sizes. Indeed, comparing Figure 18.4a and Figure 18.4c, we observe that Simultaneous Chinese Remainder Problem instances with larger remainder sets tend to have smaller minimal solutions when compared to Simultaneous Chinese Remainder Problem instances with smaller remainder sets. In particular, it seems that for Simultaneous Chinese Remainder Problem instances with large remainder sets a simple brute force counting method is rather efficient. However, the same does not hold for small remainder set sizes. In this logic, the "hardest" Simultaneous Chinese Remainder Problem instances are those with remainder sets of size two, which is aligned with the conclusion in Section 10.2.9.

Let us concentrate on the special case $|\mathcal{R}_i| = 2$ for all $i \in \{1, \dots, k\}$. By Equation (18.4), there are $|R(k)| = \prod_{i=1}^{k} \binom{m_i}{2} = \frac{1}{2^k} \prod_{i=1}^{k} m_i(m_i - 1)$ Simultaneous Chinese Remainder Problem instances with remainder sets of size two. By Equation (18.6), there are $|S(0)| = \prod_{i=1}^{k} \binom{m_i - 1}{1} = \prod_{i=1}^{k} (m_i - 1)$ Simultaneous Chinese Remainder Problem instances with minimal solution 0, such that the probability of sampling such an instance uniformly at random from $R(k)$ is $\prod_{i=1}^{k} \frac{2}{m_i}$. The simulating geometric distribution from Section 18.3 is then defined by $p = \prod_{i=1}^{k} \frac{2}{m_i}$ and has mean $\mu = (\prod_{i=1}^{k} \frac{m_i}{2}) - 1$ and median $q_{1/2} \leq \log(2) \prod_{i=1}^{k} \frac{m_i}{2}$. In particular, this predicts that the majority of Simultaneous Chinese Remainder Problem instances with remainder set size two have a rather small minimal solution, however the tail of the distribution is rather long. Concretely, Section 18.4 yields that the maximal minimal solution of Simultaneous Chinese Remainder Problem instances with remainder set size two is lower bounded by $\mathbf{M} = M - 1 - \sum_{i=1}^{k} M_i$, which in turn is bounded by

$$M_1(m_1 - 1 - k) \leq \mathbf{M} \leq M_k(m_k - 1 - k). \tag{18.71}$$

If $m_1 \geq k + 1$, then the maximal minimal solution is larger than M_1 and so a brute-force counting method is generally not recommended to solve this kind of Simultaneous Chinese Remainder Problem instances. This confirms our intuition on the hardness of the Simultaneous Chinese Remainder Problem instances with small remainder set sizes.



(a) SimCRP with 3 moduli

(b) SimCRP with 6 moduli

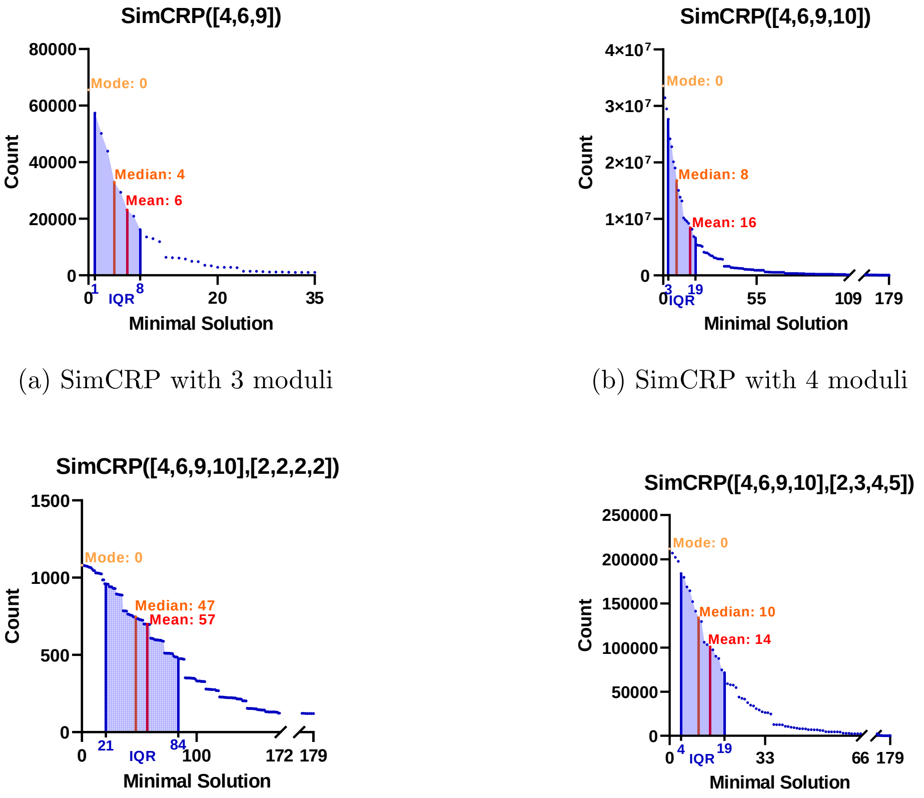
(c) SimCRP with 3 moduli

(d) SimCRP with 3 moduli

Figure 18.5: Scatter plots for the minimal solution of Simultaneous Chinese Remainder Problem instances with fixed moduli and remainder sets of size 2 (blue) featuring a simulation using the geometric distribution (black). An axis cut has been made at 99% of the counts. The mean is rounded to the closest integer and the rightmost entry on the horizontal axis represents the maximal minimal solution.

18.6 Moduli with common divisors

Our statistics were developed for pairwise coprime moduli only. This choice is due to our focus on the Minimal Chinese Remainder Problem that is defined for pairwise coprime moduli. As Figure 18.6 shows, composite moduli produce a similar distribution, but with some irregularities such as the range of minimal solutions. The number of instances with no solution is discarded from the distributions.



(a) SimCRP with 3 moduli

(b) SimCRP with 4 moduli

(c) SimCRP with 4 moduli and remainder sets of size 2

(d) SimCRP with 4 moduli and remainder sets of varying size

Figure 18.6: Scatter plots for the minimal solution of Simultaneous Chinese Remainder Problem instances with fixed non-coprime moduli. Except for (a), an axis cut has been made at 99% of the counts. The mean is rounded to the closest integer and the rightmost entry on the horizontal axis represents the maximal minimal solution.

Chapter 19

Open Questions

The Simultaneous Chinese Remainder Problem remains relatively unexplored and many open questions on the topic persist.

The intriguing reduction of 3-SAT to the Existential Simultaneous Chinese Remainder Problem in Section 10.1 enables an elementary number theoretic framework for the study of a well-known complexity-theoretic problem. Advances in one direction may have a direct impact on the other one. The same holds for the reduction constructed in Section 10.2 for the Bounded Simultaneous Chinese Remainder Problem. The empirical observations in Chapter 18, indicate that most Simultaneous Chinese Remainder Problem instances have a rather small minimal solution, even for small remainder sets. Thereby, it might be interesting to attempt a classification of "easy" problems and to investigate the corresponding 3-SAT instances. The challenge in such an analysis lies in outlining the properties of Simultaneous Chinese Remainder Problems obtained from the given reductions as they cannot be seen as randomly sampled.

Concerning our statistical experiments, we note that a computational verification of our observations in Chapter 18 with sufficiently many large moduli may reveal new properties of the minimal solutions. A full description of the maximal minimal solution of a Simultaneous Chinese Remainder Problem with fixed remainder set sizes would improve the rough bound from Chapter 12. Furthermore, the development of an upper bound that does not only depend on the remainder set sizes, but also on the internal structure of the remainder sets may help us better understand the general problem.

The compelling "modulus switch" described in Chapter 14 raises the question of the internal structure of Simultaneous Chinese Remainder Problem solutions. Mixed-radix comparison methods may be used to improve the investigated solving techniques or to give a new perspective on solving techniques. For example, the described lattice constructions to find the minimal solution may be improved. Regarding lattices, we may also formalize a new Simultaneous Chinese Remainder Problem asking to find the maximal distance between any two solutions for a given instance. This corresponds to the *Bounded Distance Decoding Problem* and seems *a priory* as hard as the other variants that we studied. Finally, the construction of a special purpose fully polynomial-time approximation scheme for finding the minimal solution would simultaneously shed light on the “hard” instances and reveal new bounds for the maximal minimum.

Bibliography II

[BN00] Daniel Bleichenbacher and Phong Q. Nguyen. Noisy polynomial interpolation and noisy Chinese remaindering. In *Advances in Cryptology—EUROCRYPT 2000 (Bruges)*, volume 1807 of *Lecture Notes in Comput. Sci.*, pages 53–69. Springer, Berlin, 2000.

[Bru15] Viggo Brun. Über das Goldbachsche Gesetz und die Anzahl der Primzahlpaare. *Archiv fur Mathematik und Naturvidenskab*, B34, 1915.

[CM05] Alina Carmen Cojocaru and M. Ram Murty. *An Introduction to Sieve Methods and Their Applications.* London Mathematical Society Student Texts. Cambridge University Press, 2005.

[Coh93] Henri Cohen. *A Course in Computational Algebraic Number Theory*, volume 138 of *Graduate Texts in Mathematics.* Springer-Verlag, Berlin, 1993.

[Cop97] Don Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

[DIP93] Giovanni Dimauro, Sebastiano Impedovo, and Giuseppe Pirlo. A new technique for fast number comparison in the residue number system. *IEEE Transactions on Computers*, 42(5):608–612, 1993.

[DPS96] Cunsheng Ding, Dingyi Pei, and Arto Salomaa. *Chinese Remainder Theorem: Applications in Computing, Coding, Cryptography.* World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996.

[Dus99] Pierre Dusart. The kth prime is greater than k(ln k + ln ln k - 1) for k $\geq$ 2. *Mathematics of Computation*, 68(225):411–415, 1999.

[FH96] David Ford and George Havas. A new algorithm and refined bounds for extended gcd computation. In *Proceedings of the Second International Symposium on Algorithmic Number Theory*, ANTS-II, page 145–150, Berlin, Heidelberg, 1996. Springer.

[For20] Kevin Ford. Sieve Methods Lecture Notes. https://faculty.math.illinois.edu/~ford/sieve2020.pdf, Spring 2020.

[Gal12] Steven D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, USA, 1st edition, 2012.

[Gar59] Harvey L. Garner. The residue number system. In *Papers Presented at the the March 3-5, 1959, Western Joint Computer Conference*, IRE-AIEE-ACM '59 (Western), page 146–153, New York, NY, USA, 1959. Association for Computing Machinery.

[Gau95] Carl Friedrich Gauss. *Disquisitiones Arithmeticae*, volume 10 of *Colección Enrique Pérez Arbeláez [Enrique Pérez Arbeláez Collection]*. Academia Colombiana de Ciencias Exactas, Físicas y Naturales, Bogotá, 1995. Translated from the Latin by Hugo Barrantes Campos, Michael Josephy and Ángel Ruiz Zúñiga, With a preface by Ruiz Zúñiga.

[GG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3 edition, 2013.

[Has36] Helmut Hasse. Zur Theorie der abstrakten elliptischen Funktionenkörper. I: Die Struktur der Gruppe der Divisorenklassen endlicher Ordnung. *J. Reine Angew. Math.*, 175:55–62, 1936.

[HH11] Hans-Egon Richert Heini Halberstam. *Sieve Methods*. Dover Publications, Mineola, New York, 2011.

[HKZZ19] Thomas Dueholm Hansen, Haim Kaplan, Or Zamir, and Uri Zwick. Faster k-SAT algorithms using biased-PPSZ. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 578–589, New York, NY, USA, 2019. Association for Computing Machinery.

[HLLP16] Cheng He, Joseph Y.T. Leung, Kangbok Lee, and Michael L. Pinedo. An improved binary search algorithm for the multiple-choice knapsack problem. *RAIRO - Operations Research*, 50(4-5):995–1001, October 2016. Publisher Copyright: © 2016 EDP Sciences, ROADEF, SMAI.

[HvdH21] David Harvey and Joris van der Hoeven. Integer multiplication in time $O(n\log n)$. *Annals of Mathematics*, 193(2):563 – 617, 2021.

[Isu16] Konstantin Isupov. An algorithm for magnitude comparison in RNS based on mixed-radix conversion II. *International Journal of Computer Applications*, 141:1–4, 05 2016.

[Kat93] Victor J. Katz. *A History of Mathematics*. HarperCollins College Publishers, New York, 1993. An introduction.

[KPP04] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *The Multiple-Choice Knapsack Problem*, pages 317–347. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[Law77] E. L. Lawler. Fast approximation algorithms for knapsack problems. In *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*, pages 206–213, 1977.

[Lib73] Ulrich Libbrecht. *Chinese Mathematics in the Thirteenth Century*. M.I.T. Press, Cambridge, 1973. The Shu-shu Chiu-chang of Ch'in Chiu-shao, MIT East Asian Science Series, 1.

[Lip71] John D. Lipson. Chinese remainder and interpolation algorithms. *Proceedings of the Second ACM Symposium on Symbolic and Algebraic Manipulation*, page 372–391, 1971.

[Lip09] Richard J. Lipton. Gödel's Lost Letter and P=NP. https://rjlipton.wpcomstaging.com/2009/08/01/the-chinese-remainder-theorem-with-limits/, 2009. Last accessed 14.04.2021.

[MA76] Kenneth Manders and Leonard Adleman. NP-complete decision problems for quadratic polynomials. In *Proceedings of the Eighth Annual ACM Symposium on Theory of Computing*, STOC '76, page 23–29, New York, NY, USA, 1976. Association for Computing Machinery.

[Mac82] Allan J Macleod. A comparison of algorithms for polynomial interpolation. *Journal of Computational and Applied Mathematics*, 8(4):275 – 277, 1982.

[Mö08] Niels Möller. On Schönhage's algorithm and subquadratic integer GCD computation. *Math. Comput.*, 77:589–607, 07 2008.

[Pis03] David Pisinger. Dynamic programming on the word RAM. *Algorithmica*, 35:128–145, 2003.

[RSA78] Ron L. Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, feb 1978.

[RZ04] Fabrice Rouillier and Paul Zimmermann. Efficient isolation of polynomial's real roots. *Journal of Computational and Applied Mathematics*, 162(1):33–50, 2004. Proceedings of the International Conference on Linear Algebra and Arithmetic 2001.

[Sch87] Isaac Schoenberg. The Chinese remainder problem and polynomial interpolation. *The College Mathematics Journal*, 18:320, 09 1987.

[Sch95] René Schoof. Counting points on elliptic curves over finite fields. *Journal de Théorie des Nombres de Bordeaux*, 7(1):219–254, 1995.

[Sor98] Jonathan Sorenson. Trading time for space in prime number sieves. *ANTS*, 1423, 05 1998.

[ST67] Nicholas S. Szabo and Richard I. Tanaka. *Residue Arithmetic and Its Application to Computer Technology*. McGraw-Hill, New York, NY, USA, 1967.

[VDH16] Joris Van Der Hoeven. Faster Chinese Remaindering. https://hal.archives-ouvertes.fr/hal-01403810, November 2016. Last accessed 05.05.2022.

[Xia99] Xiang-Gen Xia. On estimation of multiple frequencies in undersampled complex valued waveforms. *IEEE Transactions on Signal Processing*, 47(12):3417–3419, Dec 1999.

[Xia00] Xiang-Gen Xia. An efficient frequency-determination algorithm from multiple undersampled waveforms. *Signal Processing Letters, IEEE*, 7:34 – 37, 03 2000.

[ZX97] Guangcai Zhou and Xiang-Gen Xia. Multiple frequency detection in undersampled complex-valued waveforms with close multiple frequencies. *Electronics Letters*, 33(15):1294–1295, July 1997.

Part III

A Conjecture On Primes In Arithmetic Progressions And Geometric Intervals

The contents of this part are going to appear as:

Jim Barthel and Volker Müller. (2022). A Conjecture on Primes in Arithmetic Progressions and Geometric Intervals, *Amer. Math. Monthly*

Act III: The second count

Despite the similarity to the traditional remainder computation, the uncertainty of the remainders seemed to make the problem exponentially harder. Jay's pa confessed: “I found a list of potential candidates for the number of pennies that you counted the day before, but there is no possibility to find the exact number without additional information.” Pops confirmed: “Unfortunately, the total number of pennies from last year's count does not sufficiently reduce the list of candidates and as my memory leaves much to be desired, I don't remember the number of new acquisitions”. “It seems that you need to recount the whole lot”, announced Jay's pa.

Jay was devastated. He looked at the pile of pennies and thought, “this will take a while”. He had calluses on his fingers from the first count but started nonetheless with the second count. He moved the cold metal to the middle of the table and prepared another page in his notebook. As pops knew that Jay intended to play with Missy, he wanted to help him. However, he knew that Jay needed to complete this task on his own. Instead of helping to count, he gave Jay some advice: ”If you make sure that the count values cannot be divided by the same number, then you need fewer counts”. Jay was thrilled as he believed that not even his pa knew this secret.

“How should I choose the count values”, asked Jay curiously. “You will find a way”, replied pops, “remember the prime numbers that you learned in school?”. Having in mind the disaster created by stupid Mrs Skizzles, Jay wanted to avoid that his work could be destroyed anew. He guessed that if all the count values had the same form, then he could remember them by heart. As his birthday is the fourth of July, he looked for primes that are a multiple of seven plus four. He quickly found 11, but then needed to count to 53. He was surprised that there was one prime between 7 and 7^2 and another between 7^2 and 7^3. “Is the same true if I choose other numbers”, he asked himself, “and what happens for larger exponents?”

Abstract III

Dirichlet's theorem on primes in arithmetic progressions states that for any positive integer q and any coprime integer a, there are infinitely many primes in the arithmetic progression $a + nq$ ($n \in \mathbb{N}$). The *Prime Number Theorem for arithmetic progressions* is slightly more precise and outlines asymptotic intervals where those primes can be found. However, neither of both results predicts the exact location of primes in arithmetic progressions so that their distribution needs yet to be determined. A particular interest lies in the first prime p_0 that can be found in an arithmetic progression $a + nq$. *Linnik's theorem* gives the elegant upper bound $p_0 \leq q^L$ where L denotes an absolute and explicitly computable constant. Albeit only $L = 5$ has been proven, it is widely believed that $L \leq 2$. Hereinafter, we postulate the following explicit generalization of *Linnik*'s theorem:

Conjecture. *For any integers $q \geq 2$, $1 \leq a \leq q - 1$ with $\gcd(q, a) = 1$, and $t \geq 1$, there exists a prime p such that*

$$q^t \leq p < q^{t+1} \quad \text{and} \quad p \equiv a \mod q.$$

Given today's best upper bound of *Linnik's constant* L, the proof of this conjecture is likely to be out of reach. Nonetheless, the conjecture can be proven for all sufficiently large t and computationally verified for all sufficiently small q. Surprisingly, the conjecture has a direct impact on a claim of *Pomerance* related to *Carmichael's totient conjecture*, both results being thoroughly discussed herein.

Contents III

Act III **163**

Abstract III **164**

Contents III **165**

20 The distribution of primes **167**

20.1 Primes in general 167
20.2 Primes in arithmetic progressions 168
20.3 Another prime counting function 169
20.4 A generalization of primes in arithmetic progressions 170

21 A new conjecture **171**

21.1 Linnik's constant 171
21.2 A generalization of Linnik's constant 172

22 A partial proof **173**

22.1 An elementary proof for progression size q=2 173
22.2 An existential proof for every sufficiently large exponent t . . 173
22.3 Consideration under ERH 175
22.4 Computational verification 177
22.5 Computational cost 178
22.6 Conclusion 179

23 Relation to other conjectures **181**

23.1 Carmichael's conjecture 181
23.2 Computational verification 181
23.3 Pomerance's conjecture 182
23.4 The link with our conjecture 183

24 Open Questions **185**

Bibliography III **187**

Chapter 20

The distribution of primes

20.1 Primes in general

Circa 300. BC in Book IX Proposition 20 of his famous *Elements* [Wil88], *Euclid* delivered the first proof for the infinity of primes. Only 21 centuries later the claim was made more precise by *Legendre* [Leg09] (and *Gauss* [Gau76]) approximating the number of primes $\pi(x)$ below a given bound x. In 1896 the conjectured asymptotic behaviour

$$\pi(x) \sim \frac{x}{\log(x)} \tag{20.1}$$

got independently proven by *Hadamard* [Had93, Had96] and *De la Vallée Poussin* [DlVP96]. Their result is known as the *Prime Number Theorem* and has been improved considerably so that more precise estimates are known today. *Dusart* [Dus98] proved that

$$\pi(x) = \frac{x}{\log(x)}\left(1 + \frac{1}{\log(x)} + \frac{2}{\log^2(x)} + O\left(\frac{1}{\log^3(x)}\right)\right), \tag{20.2}$$

and *Trudigan* [Tru14] showed that for all $x \geq 229$

$$|\pi(x) - li(x)| \leq 0.2795\,\frac{x}{\log^{3/4}(x)} e^{-\sqrt{\frac{\log(x)}{6.455}}} \tag{20.3}$$

where li denotes the logarithmic integral $li(x) := \int_0^x \frac{dt}{\log(t)}$. An extensive review of similar results can be found in [Kou19].

20.2 Primes in arithmetic progressions

In the same line of work, primes with particular properties were studied. A specific interest laid on primes in arithmetic progressions. *Dirichlet's theorem on primes in arithmetic progressions* [Dir37] guarantees that there are infinitely many primes in the arithmetic progression $a + nq$ $(n \in \mathbb{N})$ if the starting point a and the progression size q are coprime. *De la Vallée Poussin* [DlVP96] outlined that primes are evenly distributed among the coprime congruence classes of q. His *Prime Number Theorem for arithmetic progressions* yields that

$$\pi(x; q, a) \sim \frac{x}{\varphi(q)\log(x)} \tag{20.4}$$

where $\pi(x; q, a)$ denotes the number of primes in the arithmetic progression $a+nq$ smaller than x and φ denotes the *Euler totient function* [Gau66, §38].

Using *Brun's sieve* [Bru15], *Titchmarsh* [Tit30] proved that for any $0 < a < q$ such that $\gcd(a, q) = 1$ and any $x > q$,

$$\pi(x; q, a) = O\left(\frac{1}{1 - \frac{\log(q)}{\log(x)}} \frac{x}{\varphi(q)\log(x)}\right) \tag{20.5}$$

where the implied constant is effectively computable. *Montgomery* and *Vaughan* [MV73] managed to upper bound the implied constant and deduced that

$$\pi(x; q, a) < \frac{2x}{\varphi(q)\log(\frac{x}{q})}. \tag{20.6}$$

Furthermore, their construction shows that the same bound holds for any interval of length x, that is, for any $y > 0$,

$$\pi(y + x, q, a) - \pi(y, q, a) < \frac{2x}{\varphi(q)\log(\frac{x}{q})}. \tag{20.7}$$

If x is sufficiently large, these bounds can be further improved. For example, *Maynard* [May13] developed an estimation of *Dirichlet L-functions* [Dir37], in which he dealt with exceptional zeroes and applied the *Deuring-Heilbronn phenomenon* [Deu33, Hei34], to deduce that for all $x > q^8$ the upper bound

$$\pi(x; q, a) < \frac{2}{\varphi(q)} Li(x) \tag{20.8}$$

holds, where $Li(x) := li(x) - li(2)$. Furthermore, under the same conditions, he deduced the asymptotic behaviour

$$\pi(x; q, a) = O\left(\frac{\log(q)}{q^{1/2}} \frac{x}{\varphi(q)\log x}\right), \tag{20.9}$$

where the implied constant is effectively computable.

20.3 Another prime counting function

Despite its remarkably simple definition, the natural prime counting function

$$\pi(x; q, a) = \sum_{\substack{p \le x \\ p \equiv a \mod q}} 1 \tag{20.10}$$

is generally not convenient to work with. Indeed, as a function by constant steps, it is rather difficult to approach by a continuous function and smoothening it typically implies a large error. Thus, to allow for elementary analytic transformations, we may consider functions that add a non-constant term for each prime. The *first Tchebychev prime counting function for arithmetic progressions* is defined by

$$\theta(x; q, a) := \sum_{\substack{p \le x \\ p \equiv a \mod q}} \log(p). \tag{20.11}$$

Through a meticulous analysis of *Dirichlet L-functions* and an extensive computational study, *Bennett et al.* [BMOR18] achieved the following tight bounds for all sufficiently small progression sizes.

Theorem 20.1 (Bennett, Martin, O'Bryant, Rechnitzer, [BMOR18])**.** *Let $q \ge 3$ and $0 < a < q$ be integers such that $\gcd(a, q) = 1$. Then, there exist positive constants $c_\theta(q)$ and $x_\theta(q)$ such that*

$$\left|\theta(x; q, a) - \frac{x}{\varphi(q)}\right| < c_\theta(q) \frac{x}{\log(x)}$$

for all $x \ge x_\theta(q)$. Moreover, $c_\theta(q) \le c_0(q)$ and $x_\theta(q) \le x_0(q)$ where

$$c_0(q) = \begin{cases} \frac{1}{840} & \textit{if} \quad 3 \le q \le 10^4, \\ \frac{1}{160} & \textit{if} \quad 10^4 < q, \end{cases}$$

and

$$x_0(q) = \begin{cases} 8 \cdot 10^9 & \textit{if} \quad 3 \le q \le 10^5, \\ e^{0.03\sqrt{q}(\log(k))^3} & \textit{if} \quad 10^5 < q. \end{cases}$$

A complete list of optimal constants c_θ and x_θ for all $3 \leq q \leq 10^5$ and their enormous underlying data set is available at

https://www.nt.math.ubc.ca/BeMaObRe/.

20.4 A generalization of primes in arithmetic progressions

Chebotarev's density theorem [Tsc26] is a generalization of the *Prime Number Theorem for arithmetic progressions*. To be precise, let L, F be number fields and let L/F be a Galois extension with Galois group $G := Gal(L/F)$. For a prime ideal $\mathfrak{p}$ of F that is unramified in L, let $\left[\frac{L/F}{\mathfrak{p}}\right]$ be the conjugacy class of Frobenius automorphisms in G above $\mathfrak{p}$. Then, *Chebotarev's density theorem* yields that for any $x > 1$ and any conjugacy class $C \subseteq G$,

$$\pi(x; C, L/F) \sim \frac{|C|}{|G|} \frac{x}{\log(x)} \tag{20.12}$$

where $\pi(x; C, L/F)$ denotes the number of prime ideals of F that are unramified in L such that $\left[\frac{L/F}{\mathfrak{p}}\right] = C$ and the absolute norm $N_{F/\mathbb{Q}}(\mathfrak{p})$ is smaller than x. Choosing $F = \mathbb{Q}$ and $L = \mathbb{Q}\left[e^{\frac{2\pi i}{q}}\right]$ yields the *Prime Number Theorem for arithmetic progressions*. Albeit explicit estimates of the implied error term exist (e.g., see *Lagarias*' and *Odlyzko*'s estimates in [LO77] and its subsequent improvements), the tightness of the resulting bounds cannot compete with the precision of the current error estimates for the *Prime Number Theorem for arithmetic progressions*. A detailed account of this conclusion and related information can be found in [Zam17].

Chapter 21

A new conjecture

The study of the distribution of primes in fixed intervals turns out to be highly challenging. Whereas cumulative results are usually obtained by smoothening the prime counting functions or by a convenient sieving method, both approaches are inefficient if short intervals are involved because the committed error becomes too important.

21.1 Linnik's constant

The best illustration of the complexity of studying the distribution of primes with particular properties in short intervals is the strenuous quest for predicting the size of the smallest prime in an arithmetic progression. A milestone has been achieved by *Linnik* [Lin44a, Lin44b] in 1944 with the proof that there is an absolute upper bound polynomial in the progression size.

Theorem 21.1 (Linnik)**.** *There are absolute constants C and L such that for any integer $q \geq 2$ and any integer $1 \leq a \leq q-1$ with $\gcd(a,q)=1$, the smallest prime $p_0 \equiv a \mod q$ satisfies $p_0 \leq Cq^L$.*

Although Linnik's original development did not contain estimates for his predicted constants C and L, his development showed them to be effectively computable. The infimum over all possible absolute constants L for which Theorem 21.1 holds is known as *Linnik's constant.* A stream of work, outlined in *Heath-Brown*'s seminal work [HB92], developed a decreasing sequence of upper bounds for L leading to today's best unconditional upper bound $L \leq 5$ [Xyl11]. Under the *Generalized Riemann Hypothesis* (GRH), *Chowla* [Cho34] concludes that any constant $L > 2$ is admissible. Furthermore, aligned with *Schinzel's conjecture* H_2 [SS58], *Heath-Brown* [HB92] conjectures the unconditional upper bound $L \leq 2$, and *Bombieri et al.* [BFI89] show that this claim holds for almost all moduli.

21.2 A generalization of Linnik's constant

Leaning on the discussion about the smallest prime in an arithmetic progression, the question on the position of subsequent primes arises. By keeping the coefficient $C = 1$ constant in Theorem 21.1 and allowing the exponent L to vary, the positions of these primes can be roughly bounded.

Conjecture 21.2 (Barthel, Müller [BM22]). *For any integers $q \geq 2$, $1 \leq a \leq q-1$ with $\gcd(q,a) = 1$ and $t \geq 1$, there exists a prime p such that*

$$q^t \leq p < q^{t+1} \quad \textit{and} \quad p \equiv a \mod q.$$

Considering $t = 1$ and dropping the lower bound for the desired prime p, we recover the conjectured size $L \leq 2$ of *Linnik's constant*. Consequently, it is unlikely to find a full proof of Conjecture 21.2 with the current state of mathematics. Nonetheless, as the geometric intervals $[q^t, q^{t+1})$ expand faster than the occurrence of primes decreases, a proof can be anticipated for sufficiently large t. Based on the asymptotic behaviour of the Prime Number Theorem for arithmetic progressions, we can even shrink the considered intervals for large t and still prove the existence of such primes. Whereas the approximation of the error term in the *Prime Number Theorem for arithmetic progressions* is a prominent research direction, its main results, like Equation (20.5), describe the asymptotic behaviour of the prime counting functions only which usually hides large implicit constants. Only a few explicit results, such as Theorem 20.1, exist. Conjecture 21.2 yields a practical range for the positions of primes in arithmetic progressions without any hidden constant or complicated coefficients. The trade-off for its elementary description is the optimality of its range.

Chapter 22

A partial proof

22.1 An elementary proof for progression size q=2

If the progression size in Conjecture 21.2 is $q = 2$, then the conjecture claims that for any exponent $t \geq 1$, there exists a prime p such that

$$2^t \leq p < 2^{t+1} \quad \text{and} \quad p \equiv 1 \mod 2. \tag{22.1}$$

As any prime, other than 2, is odd, the claim follows directly from *Bertrand's postulate* [Ber45] affirming that for any $n \geq 2$, there is a prime p such that $n < p < 2n$. Initially proven by *Tchebychev* [Tch52] through an analytic development, *Bertrand's postulate* follows also from *Sylvester's theorem* [Syl12], a combinatorial result stating that the product of k consecutive integers strictly larger than k is necessarily divisible by a prime greater than k. Although, *Laishram*, *Shorey*, and *Tijdeman* [LS06, ST07] generalized *Sylvester's theorem* to consecutive elements in arithmetic progressions, their results are not sufficient to conclude the existence of a prime in the desired range $[q^t, q^{t+1})$ for $q \geq 3$.

22.2 An existential proof for every sufficiently large exponent t

A proof for larger moduli can be partially obtained by analytical methods. As described at the end of Chapter 21, we may use the Prime Number Theorem for arithmetic progressions to conclude Conjecture 21.2 for every sufficiently large exponent t. However, for the sake of explicit results, we rely on Theorem 20.1 only. More precisely, using the explicit constants of

Theorem 20.1, we devise

$$T_\theta(q) := \max\left\{\frac{c_\theta(q)\varphi(q)(q+2)}{(q-1)\log(q)} - 1;\ \log_q(x_\theta(q)); 1\right\}, \tag{22.2}$$

which represents an explicit lower bound for the exponents t verifying Conjecture 21.2.

Lemma 22.1. *Let $q \geq 3$, $1 \leq a \leq q-1$ with $\gcd(a,q) = 1$, and $t \geq T_\theta(q)$. Then, there exists a prime p such that $q^t \leq p < q^{t+1}$ and $p \equiv a \mod q$.*

Proof. By construction, $t \geq T_\theta(q) \geq \log_q(x_\theta(q))$, and so

$$q^{t+1} > q^t \geq q^{T_\theta(q)} \geq q^{\log_q(x_\theta(q))} = x_\theta(q). \tag{22.3}$$

Thus, Theorem 20.1 yields the two estimates

$$\frac{q^{t+1}}{\varphi(q)} - c_\theta(q)\frac{q^{t+1}}{(t+1)\log(q)} < \theta(q^{t+1}; q, a) < \frac{q^{t+1}}{\varphi(q)} + c_\theta(q)\frac{q^{t+1}}{(t+1)\log(q)} \tag{22.4}$$

and

$$\frac{q^t}{\varphi(q)} - c_\theta(q)\frac{q^t}{t\log(q)} < \theta\left(q^t; q, a\right) < \frac{q^t}{\varphi(q)} + c_\theta(q)\frac{q^t}{t\log(q)}. \tag{22.5}$$

First, subtracting the upper bound of Equation (22.5) from the lower bound of Equation (22.4) and subsequently using $t \geq T_\theta(q) \geq 1$ gives

$$\theta\left(q^{t+1}; q, a\right) - \theta\left(q^t; q, a\right) > q^t\left[\frac{q-1}{\varphi(q)} - \frac{c_\theta(q)}{\log(q)}\frac{qt+t+1}{t(t+1)}\right], \tag{22.6}$$

$$\geq q^t\left[\frac{q-1}{\varphi(q)} - \frac{c_\theta(q)}{\log(q)}\frac{q+2}{t+1}\right]. \tag{22.7}$$

By construction, $t \geq T_\theta(q) \geq \frac{c_\theta(q)\varphi(q)(q+2)}{(q-1)\log(q)} - 1$, and so the last quantity is non-negative. Thus, $\theta\left(q^{t+1}; q, a\right) > \theta\left(q^t; q, a\right)$ which guarantees the existence of a prime $p \in [q^t, q^{t+1})$ such that $p \equiv a \mod q$. □

As one expects, the lower bound on the exponent t in Lemma 22.1 depends only on the modulus q and not on the congruence class a.

Remark 22.2. *A direct computation using the constants x_θ and c_θ shows that for all $3 \leq q \leq 45000$ we have*

$$T_\theta(q) > 1.$$

Thus $\lceil T_\theta(q)\rceil \geq 2$, which is required in Section 22.4.

Remark 22.3. *If $T_\theta(q) > 8$, a tighter lower bound for t can be achieved by considering Maynard's estimate in Equation (20.8) for the upper bound of $\theta\left(q^{t+1}; q, a\right)$.*

22.3 Consideration under ERH

Assuming the *Extended Riemann Hypothesis* (ERH), we can reduce the lower bound on the exponents t for which Conjecture 21.2 holds. Indeed, under ERH, the error term in the *Prime Number Theorem for arithmetic progressions* can be drastically decreased. *Bach* and *Shallit* [BS96] developed the following explicit estimate.

Theorem 22.4 (Bach, Shallit)**.** *Let $q \geq 3$ be an integer and let a be an integer that is coprime to q. Then, assuming ERH,*

$$\left| \pi\left(x; q, a\right) - \frac{li(x)}{\varphi(q)} \right| < \sqrt{x}(\log(x) + 2\log(q)) \qquad \forall x \geq 2.$$

Subtracting the upper bound for $\pi\left(q^t; q, a\right)$ from the lower bound for $\pi\left(q^{t+1}; q, a\right)$ yields

$$\begin{aligned} &\pi\left(q^{t+1}; q, a\right) - \pi\left(q^t; q, a\right) \\ &> \frac{li\left(q^{t+1}\right) - li\left(q^t\right)}{\varphi(q)} - \sqrt{q^t}\log(q)(\sqrt{q}(t+3) + (t+2)). \end{aligned} \tag{22.8}$$

Repeated integration by parts reveals the following recursive approximation of the difference of logarithmic integrals.

Lemma 22.5. *For all $\alpha > \beta > 1$ and all $n \in \mathbb{N} \setminus \{0\}$, we have*

$$li\left(\alpha\right) - li\left(\beta\right) = \left(\sum_{i=1}^{n} (i-1)! \frac{\alpha \log\left(\beta\right)^i - \beta \log\left(\alpha\right)^i}{\log\left(\alpha\right)^i \log\left(\beta\right)^i} \right) + n! \int_{\beta}^{\alpha} \frac{dt}{\log(t)^{n+1}}.$$

As $\frac{1}{\log(t)^{n+1}} > 0$ for all $t > 1$, the integral $\int_\beta^\alpha \frac{dt}{\log(t)^{n+1}}$ is strictly positive for all $\alpha > \beta > 1$ which implies the following approximation.

Lemma 22.6. *For all $\alpha > \beta > 1$ and all $n \in \mathbb{N} \setminus \{0\}$, we have*

$$li\left(\alpha\right) - li\left(\beta\right) > \sum_{i=1}^{n} (i-1)! \frac{\alpha \log\left(\beta\right)^i - \beta \log\left(\alpha\right)^i}{\log\left(\alpha\right)^i \log\left(\beta\right)^i}.$$

Choosing $n = 1$, $\alpha = q^{t+1}$ and $\beta = q^t$ in Lemma 22.6, we find the lower bound

$$li\left(q^{t+1}\right) - li\left(q^t\right) > \frac{q^t}{\log(q)} \left(\frac{qt - (t+1)}{t(t+1)} \right). \tag{22.9}$$

Inserting this lower bound into (22.8) yields

$$\begin{aligned}&\pi\left(q^{t+1};q,a\right)-\pi\left(q^{t};q,a\right)\\&\quad>\frac{q^t(qt-(t+1))}{\varphi(q)\log(q)t(t+1)}-\sqrt{q^t}\log(q)(\sqrt{q}(t+3)+(t+2))\end{aligned}\tag{22.10}$$

and using the trivial bound $\varphi(x)\le x$ gives

$$\begin{aligned}&\pi\left(q^{t+1};q,a\right)-\pi\left(q^{t};q,a\right)\\&\quad>\frac{q^{t-1}(qt-(t+1))}{\log(q)t(t+1)}-\sqrt{q^t}\log(q)(\sqrt{q}(t+3)+(t+2)).\end{aligned}\tag{22.11}$$

If $t\ge 4$, then we observe that for all $q\ge 36$, we have

$$\sqrt{q}(t+3)+(t+2)\le 2t\sqrt{q}\tag{22.12}$$

and so

$$\pi\left(q^{t+1};q,a\right)-\pi\left(q^{t};q,a\right)>\frac{q^{t-1}(qt-(t+1))}{\log(q)t(t+1)}-2\sqrt{q^{t+1}}\log(q)t.\tag{22.13}$$

Thus,

$$\begin{aligned}\pi(q^{t+1};q,a)-\pi(q^t;q,a)&>q^{\frac{t+1}{2}}\left(\frac{q^{\frac{t-3}{2}}(qt-(t+1))}{\log(q)t(t+1)}-2\log(q)t\right),&(22.14)\\&\ge q^{\frac{t+1}{2}}\left(\frac{q^{\frac{t-3}{2}}(q-2)}{\log(q)(t+1)}-2\log(q)t\right),&(22.15)\\&=q^{\frac{t+1}{2}}\frac{\log(q)}{t+1}\left(q^{\frac{t-3}{2}}\frac{q-2}{\log(q)^2}-2t(t+1)\right),&(22.16)\\&>q^{\frac{t+1}{2}}\frac{\log(q)}{t+1}\underbrace{\left(q^{\frac{t-3}{2}}-2t(t+1)\right)}_{=:A(q,t)},&(22.17)\end{aligned}$$

where the last inequality holds for all $q\ge 4$. An elementary function study shows that $A(q,t)\ge 0$ for all $t\ge 4$ and all $q\ge 1600$.

If $t=3$, then (22.11) yields

$$\begin{aligned}\pi\left(q^4;q,a\right)-\pi\left(q^3;q,a\right)&>\frac{q^2(3q-4)}{12\log(q)}-\sqrt{q^3}\log(q)(6\sqrt{q}+5),&(22.18)\\&\ge\frac{2q^3}{12\log(q)}-7q^2\log(q)\ =:\ B(q).&(22.19)\end{aligned}$$

where the last inequality holds for all $q \geq 25$. An elementary function study shows that $B(q) \geq 0$ for all $q \geq 2596$.

If $t = 2$, then (22.11) becomes

$$\begin{aligned}
&\pi\left(q^3; q, a\right) - \pi\left(q^2; q, a\right) \\
&> \frac{q(2q-3)}{6\log(q)} - q\log(q)(5\sqrt{q}+4), \qquad (22.20)\\
&= \frac{q^2}{3\log(q)} - \frac{q}{2\log(q)} - 5q^{\frac{3}{2}}\log(q) - 4q\log(q) =: C(q), \qquad (22.21)
\end{aligned}$$

where $C(q) \geq 0$ for all $q \geq 17386763$.

By summarizing the above development and taking care of the individual conditions on t and q, we conclude the following theorem.

Theorem 22.7. *Assume ERH, then*

1. *for all $t \geq 4$, all $q \geq 1600$, and $1 \leq a \leq q-1$ with $\gcd(q,a) = 1$,*
2. *for $t = 3$, all $q \geq 2596$, and $1 \leq a \leq q-1$ with $\gcd(q,a) = 1$,*
3. *for $t = 2$, all $q \geq 17386763$, and $1 \leq a \leq q-1$ with $\gcd(q,a) = 1$,*

there exists a prime p such that

$$q^t \leq p < q^{t+1} \quad \textit{and} \quad p \equiv a \mod q.$$

Remark 22.8. *Through an advanced analysis of the above estimates and a more rigorous development, we may further improve the lower bounds for q. For example, considering $n = 2$ terms in the approximation of $li\left(q^{t+1}\right) - li\left(q^t\right)$ in Lemma 22.6, which defines Equation (22.9), leads to the desired conclusion for $t = 2$ and all $q \geq 16484144$. For the sake of simplicity, the less precise but more readable variant has been chosen. Yet, the conjectured bound of Linnik's constant seems to be out of reach of such a development.*

22.4 Computational verification

Lemma 22.1 demonstrates unconditionally and Theorem 22.7 under ERH that for a fixed $q \geq 3$ Conjecture 21.2 holds for all sufficiently large exponents. Based on the unconditional development in Section 22.2, we know that the remaining exponents vary at most in $0 < t < T_\theta(q)$. Using the explicit lists of constants $c_\theta(q)$ and $x_\theta(q)$ from [BMOR18] published at https://www.nt.math.ubc.ca/BeMaObRe/,

we computationally verified the conjecture for those finitely many exponents for all $3 \leq q \leq 45000$.

Lemma 22.9. *Let* $3 \leq q \leq 45000$, $1 \leq a \leq q-1$ *with* $\gcd(q, a) = 1$, *and* $1 \leq t \leq T_\theta(q)$. *Then, there exists a prime* p *such that* $q^t \leq p < q^{t+1}$ *and* $p \equiv a \mod q$.

The computational verification, used a slightly optimized brute-force strategy to compute for each modulus $3 \leq q \leq 45000$ and each coprime remainder $1 \leq a \leq q-1$ an explicit list $L(q, a)$ of primes $p_1, ..., p_{\lceil T_\theta(q) \rceil - 1}$ satisfying

$$q < p_1 < q^2 < p_2 < ... < q^{\lceil T_\theta(q) \rceil - 1} < p_{\lceil T_\theta(q) \rceil - 1} < q^{\lceil T_\theta(q) \rceil} \tag{22.22}$$

and $p_i \equiv a \mod q$ for all $i \in \{1, ..., \lceil T_\theta(q) \rceil - 1\}$. We note that by construction $T_\theta(q) > 1$ (see Remark 22.2) such that the list $L(q, a)$ was non-empty. To obtain a list $L(q, a)$, the program computed for each $t \in \{1, \ldots, \lceil T_\theta(q) \rceil\}$, the first prime in the interval $[q^t, q^{t+1})$. This prime was retrieved by checking the primality of the first element in the arithmetic progression in the desired interval, namely $q^t + a$, jumping to the next element by adding q if it was composite, and repeating the same procedure until a prime was found. Prime testing of a candidate p was carried out in two steps. First, p was pre-processed by the *Baillie-PSW pseudoprime test* [BWJ80], which is known to certify primality if $p < 2^{64}$ [Fei22]. Next, upon passing this pre-processing, p was either classified a prime (if $p < 2^{64}$), or it was postprocessed by a deterministic prime certifying test (using the SageMath build-in Pari/GP *isprime* function). The resulting data sets and its SageMath source code can be found at

https://files.uni.lu/jim.barthel/PrimesInAP/.

This verification simultaneously guarantees the correctness of the conjectured size of *Linnik's constant* for all $q \leq 45000$.

22.5 Computational cost

To illustrate the time needed to carry out the computer assisted verification in Section 22.4, we highlight that for each modulus $3 \leq q \leq 45000$, the verification algorithm created $\varphi(q)$ lists $L(q, a)$, each being of size $\lceil T_\theta(q) \rceil - 1$. In particular, for a fixed prime q, the program computed one list for each

$a \in \{1, \ldots, q-1\}$. For each list, the program checked in the worst case the primality of $q^{\lceil T_\theta(q) \rceil - 1} - 2$ elements. Thus, the overall number of verifications for a fixed prime q was upper bounded by $q^{\lceil T_\theta(q) \rceil}$.

Concretely, the verification took place on an ASUS VivoBook S551LB from 2013 with an Intel Core i7-4500U dual-core (1.80GHz, 2.40GHz) processor and an 8GB HDD RAM running on Microsoft Windows 10 Home. The software used was SageMath 9.0 running on Python 3.7.3. installed in January 2020. The overall (physical) runtime was approximately 4 months, not including a second verification of correctness.

22.6 Conclusion

Combining Lemma 22.1, Theorem 22.7, and Lemma 22.9 leads to the following two conclusions.

Theorem 22.10.

1. *For all* $2 \leq q \leq 45000$*, all* $t \geq 1$*, and all* $1 \leq a \leq q-1$ *with* $\gcd(q, a) = 1$*,*

2. *for all* $q > 45000$*, all* $t \geq T_\theta(q)$*, and all* $1 \leq a \leq q-1$ *with* $\gcd(q, a) = 1$*,*

there exists a prime p such that

$$q^t \leq p < q^{t+1} \quad \textit{and} \quad p \equiv a \mod q.$$

Theorem 22.11. *Assume ERH,*

1. *for all* $2 \leq q \leq 45000$*, all* $t \geq 1$*, and all* $1 \leq a \leq q-1$ *with* $\gcd(q, a) = 1$*, and*

2. *for all* $t \geq 3$*, all* $q \geq 2$*, and all* $1 \leq a \leq q-1$ *with* $\gcd(q, a) = 1$*, and*

3. *for* $t = 2$*, all* $q \geq 17386763$*, and all* $1 \leq a \leq q-1$ *with* $\gcd(q, a) = 1$*,*

there exists a prime p such that

$$q^t \leq p < q^{t+1} \quad \textit{and} \quad p \equiv a \mod q.$$

Chapter 23

Relation to other conjectures

Conjecture 21.2 turns out to be related to some well-known open questions. Hereinafter, we outline a link to *Carmichael's totient conjecture* [Car07, Car22]. However, before exploring this relationship, we rapidly revise the known results on Carmichael's conjecture.

23.1 Carmichael's conjecture

In 1907, *Carmichael* [Car07] published a note on Euler's phi function φ claiming that the relation $\varphi(x) = n$ is never uniquely satisfied for any given value of n. However, he recognized that his proof was erroneous and he admitted that he is not in the possession of a complete proof [Car22]. Today his claim is known as *Carmichael's conjecture* and remains open.

Conjecture 23.1 (Carmichael [Car22])**.** *For a given number n, the equation $\varphi(x) = n$ either has no solution or it has at least two solutions.*

Curiously, *Ford* [For99] managed to prove that for any integer $s \geq 2$ and $s = 0$, there are infinitely many integers n such that the equality $\varphi(x) = n$ has exactly s solutions. Yet, the case $s = 1$ remains unproven.

23.2 Computational verification

Carmichael's conjecture is widely believed to be true and numerous properties of counterexamples have been discovered. Indeed, assume that there is a positive integer n such that $\varphi(x) = n$ has a unique solution x. *Carmichael* [Car22] showed that $4|x$, and if there are distinct primes $p_1, ..., p_n$ and positive integer exponents $a_1, ..., a_n$ such that $(\prod_{i=1}^{n} p_i^{a_i})|x$ and $1 + \prod_{i=1}^{n} p_i^{c_i} = P$

is a prime number for some $0 < c_i < a_i$ for all $i \in \{0, ..., k\}$, then $P^2|x$. Furthermore, he proved that if x is divisible by a *Fermat prime*, then x is divisible by the square of this prime. Subsequently, *Klee* [KJ47] generalized those observations by showing that if there are distinct primes $p_1, ..., p_n$ and positive integer exponents $a_1, ..., a_n$ such that $(\prod_{i=1}^{n} p_i^{a_i})|x$ and for two disjoint subsets $B, C \subseteq \{1, \ldots, n\}$, $1 + \left(\prod_{i\in B} p_i^{a_i-1}(p_i - 1) \prod_{j\in C} p_i^{c_i}\right) = P$ is a prime number for some $0 < c_i < a_i$ for all $i \in \{0, ..., k\}$, then $P|x$. Furthermore, if B has the property that whenever a prime $P|(p_i - 1)$ for some $i \in B$, we have $P|x$, then $P^2|x$. Additionally, he observed that it is sufficient to study counterexamples that are not divisible by 8 or any *Fermat prime* larger than 3. Using these properties, some specific optimisations, and many hours of computation time, a stream of work by *Klee*, *Schlafly*, *Wagon*, and *Ford* [KJ47, SW94, For98] managed to prove that the minimal counterexample x_0 of *Carmichael's conjecture* satisfies

$$x_0 > 10^{10^{10}}. \tag{23.1}$$

The main idea of their computational verification consists in the study of the divisor set $\mathcal{D}(x_0) = \{d \in \mathbb{N} : d|x_0\}$ of the minimal counterexample x_0 of Carmichael's conjecture. By its theoretical properties, $\{2^2, 3^2\} \subseteq \mathcal{D}(x_0)$ which then implies that $\{7^2, 43^2\} \subseteq \mathcal{D}(x_0)$. Similarly, if $3^3 \in \mathcal{D}(x_0)$, then

$$\{19^2, 127^2, 2287^2, 101347^2, 304039^2\} \subseteq \mathcal{D}(x_0). \tag{23.2}$$

Otherwise,

$$\{13^2, 79^2, 547^2, 3319^2, 1854763^2\} \subseteq \mathcal{D}(x_0). \tag{23.3}$$

In the same way, more divisors of x_0 can be computed. For a full proof in this direction, it would be sufficient to prove that x_0 has infinitely many prime factors. For example, it suffices to show that for any $\mathcal{S}_b \subseteq \mathcal{D}(x_0)$ containing all primes of $\mathcal{D}(x_0)$ up to some upper bound b, there is $\mathcal{A}' \subseteq \mathcal{A}_b$ such that $1 + \prod_{p\in\mathcal{A}'} p$ is a prime.

23.3 Pomerance's conjecture

The only known attempt for constructing a counterexample of Carmichael's conjecture has been developed by *Pomerance* [Pom74]. His construction relies on the following observation.

Theorem 23.2 (Pomerance [Pom74]). *If x is a natural number such that for every prime p, $(p - 1)|\varphi(x)$ implies $p^2|x$, then $\varphi(x) = \varphi(y)$ has only the trivial solution $y = x$.*

However, Pomerance emphasises that likely such a counterexample does not exist. Concretely, he points out that if the following conjecture holds, then there cannot be such a counterexample as it would necessarily be divisible by every single prime.

Conjecture 23.3 (Pomerance [Pom74])**.** *If $k \geq 2$, then $(p_k - 1)$ divides $\prod_{i=1}^{k-1} p_i(p_i - 1)$, where p_i denotes the i-th prime.*

23.4 The link with our conjecture

Rewriting Pomerance's conjecture in terms of primes in arithmetic progressions congruent to 1 finally reveals the link with Conjecture 21.2. Concretely, for any prime q, let $v_q : \mathbb{N} \to \mathbb{N}$ denote the q-adic valuation map defined by

$$v_q(0) = \infty \qquad \text{and} \qquad v_q(n) = \max\{v \in \mathbb{N} \colon q^v | n\}. \tag{23.4}$$

Using this notation, Pomerance's conjecture is equivalent to the claim that for all $k \in \mathbb{N}_{\geq 2}$ and all $q \in \{p_1, p_2, ..., p_{k-1}\}$,

$$v_q(p_k - 1) \leq v_q\left(\prod_{i=1}^{k-1} p_i(p_i - 1)\right). \tag{23.5}$$

Indeed, any prime divisor of $(p_k - 1)$ is smaller than p_k and so $(p_k - 1)$ divides $\prod_{i=1}^{k-1} p_i(p_i - 1)$ if and only if for all $q \in \{p_1, p_2, ..., p_{k-1}\}$ and $\ell \in \mathbb{N}$, $q^\ell | (p_k - 1)$ implies that $q^\ell | \prod_{i=1}^{k-1} p_i(p_i - 1)$. The valuation map allows for an easy comparison of those divisors by outlining for each term the highest prime power of q. Concretely, let $q \in \{p_1, p_2, ..., p_{k-1}\}$. Then

$$q^t \leq p_k - 1 < q^{t+1} \tag{23.6}$$

for some $t \in \mathbb{N}$ which trivially means that

$$v_q(p_k - 1) = t. \tag{23.7}$$

If $t = 0$, then, by the non-negativity of the valuation map, Equation (23.5) is satisfied. If $t > 0$, we need to show that $v_q\left(\prod_{i=1}^{k-1} p_i(p_i - 1)\right) \geq t$. We observe that this is a direct consequence of Conjecture 21.2. More precisely, Conjecture 21.2 predicts that for all $t' \in \{1, ..., t-1\}$ there is a prime $P_{t'}$ such that $q^{t'} \leq P_{t'} < q^{t'+1}$ and $P_{t'} \equiv 1 \mod q$; put differently, $q | (P_{t'} - 1)$.

By construction, each of these primes is smaller than p_k and belongs to $\{p_1, \ldots, p_{k-1}\}$, so that

$$\begin{aligned} v_q\left(\prod_{i=1}^{k-1} p_i(p_i-1)\right) &= 1 + v_q\left(\prod_{i=1}^{k-1}(p_i-1)\right) && (23.8)\\ &\geq 1 + v_q\left(\prod_{t'=1}^{t-1}(P_{t'}-1)\right) && (23.9)\\ &\geq 1+(t-1) = t && (23.10)\end{aligned}$$

proving Equation (23.5). Thereby, we conclude the following retroaction of Conjecture 21.2 on Carmichael's conjecture.

Theorem 23.4. *If Conjecture 21.2 holds, then Pomerance's conjecture holds, and hence there does not exist a counterexample to Carmichael's conjecture based on Theorem 23.2.*

As by Theorem 22.10, Conjecture 21.2 holds for all $q \leq 45000$, Pomerance's conjecture holds for a non-negligible proportion of all primes.

Corollary 23.5. *Pomerance's conjecture holds for any prime p_k such that the largest square factor of $p_k - 1$ is 45000-smooth.*

Proof. If $p_k - 1$ is squarefree, then, for all $q \in \{p_1, p_2, ..., p_{k-1}\}$ we have $v_q(p_k - 1) \leq 1$ and

$$v_q\left(\prod_{i=1}^{k-1} p_i(p_i-1)\right) \geq v_q\left(\prod_{i=1}^{k-1} p_i\right) \geq 1 \qquad (23.11)$$

such that Pomerance's conjecture trivially holds. Next, assume that $p_k - 1 = B^2\chi$ where B is 45000-smooth and χ is squarefree. Then, by reordering the prime factors, we obtain $p_k - 1 = B'\chi'$ where B' is 45000-smooth and χ' is either equal to 1 or squarefree with smallest prime factor being larger than 45000. In both cases, Equation (23.11) shows again that $\prod_{i=1}^{k-1} p_i(p_i - 1)$ is trivially divisible by χ'. Using the same valuation argument as above (23.6-23.10), we conclude that B' divides $\prod_{i=1}^{k-1} p_i(p_i-1)$. As B' and χ' do not share common factors, Pomerance's conjecture holds. □

Remark 23.6. *Theorem 23.4 does not contradict the general existence of a counterexample of Carmichael's conjecture, but only rules out counterexamples originating from Theorem 23.2. In this sense, Corollary 23.5 strengthens the correctness of Carmichael's conjecture but is not sufficient to validate it.*

Chapter 24

Open Questions

An unconditional proof of Conjecture 21.2 would be striking. Indeed, for $t = 1$, it would imply the conjectured size $L = 2$ of Linnik's constant. Slightly less ambitious but not less relevant is to prove the conjecture under the Extended Riemann Hypothesis. Albeit, the proof of Linnik's constant $L = 2$ is still hard under this assumption, we expect that an advanced computer-assisted development can further decrease the lower bound in Theorem 22.11 allowing to validate Conjecture 21.2 for $t = 2$ for all moduli.

As the design of the conjecture was mainly based on its elegance and practicality, one may be interested in improving the precision of the conjecture on the intervals where it holds. Explicit error estimates on the Prime Number Theorem in arithmetic progression may be used to find shorter intervals granting the existence of primes in an arithmetic progression. Furthermore, small sieves may be used to find them explicitly.

The connection of our conjecture to Pomerance's conjecture may be further exploited to study the latter claim. Albeit its impact on Carmichael's conjecture is marginal, Pomerance's conjecture is of general interest as it describes an interesting property of prime numbers.

Bibliography III

[Ber45] Joseph Bertrand. Mémoire sur le nombre de valeurs que peut prendre une fonction quand on y permute les lettres qu'elle renferme. *Journal de l'École Royale Polytechnique*, 18:123–140, 1845.

[BFI89] Enrico Bombieri, John B. Friedlander, and Henryk Iwaniec. Primes in arithmetic progressions to large moduli. III. *J. Amer. Math. Soc.*, 2(2):215–224, 1989.

[BM22] Jim Barthel and Volker Müller. A conjecture on primes in arithmetic progressions and geometric intervals. *Amer. Math. Monthly*, 2022. DOI: 10.1080/00029890.2022.2116250.

[BMOR18] Michael A. Bennett, Greg Martin, Kevin O'Bryant, and Andrew Rechnitzer. Explicit bounds for primes in arithmetic progressions. *Illinois J. Math.*, 62(1-4):427–532, 2018.

[Bru15] Viggo Brun. Über das Goldbachsche Gesetz und die Anzahl der Primzahlpaare. *Archiv for Mathematik og Naturvidenskab.*, 34:8, 1915.

[BS96] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory. Vol. 1.* Foundations of Computing Series. MIT Press, Cambridge, MA, 1996. Efficient algorithms.

[BWJ80] Robert Baillie and Samuel S. Wagstaff Jr. Lucas pseudoprimes. *Math. Comp.*, 35(152):1391–1417, 1980.

[Car07] Robert D. Carmichael. On Euler's ϕ-function. *Bull. Amer. Math. Soc.*, 13(5):241–243, 1907.

[Car22] Robert D. Carmichael. Note on Euler's φ-function. *Bull. Amer. Math. Soc.*, 28(3):109–110, 1922.

[Cho34] Sarvadaman Chowla. On the least prime in an arithmetical progression. *J. Indian Math. Soc.*, 1(2):1–3, 1934.

[Deu33] Max Deuring. Imaginäre quadratische Zahlkörper mit der Klassenzahl 1. *Math. Z.*, 37(1):405–415, 1933.

[Dir37] Lejeune Dirichlet. Beweis des Satzes, dass jede unbegrenzte arithmetische Progression, deren estes Glied und Differenz ganze Zahlen ohne gemeinschaftlichen Factor sind, unendlich viele Primzahlen enthält. *Abhandlungen der Königlich Preussischen Akademie der Wissenschaften*, pages 45–81, 1837.

[DlVP96] Charles-Jean De la Vallée Poussin. Recherches analytiques sur la théorie des nombres premiers. *Ann. Soc. Sci. Bruxelles*, 20:183—-256, 1896.

[Dus98] Pierre Dusart. *Autour de la fonction qui compte le nombre de nombres premiers*, volume 17-1998. Université de Limoges, 1998. Dissertation for the degree of Doctor of Mathematics.

[Fei22] Jan Feitsma. Pseudoprimes. http://www.janfeitsma.nl/math/psp2/index, March 16 2022.

[For98] Kevin Ford. The distribution of totients. *Ramanujan J.*, 2(1-2):67–151, 1998. Paul Erdős (1913–1996).

[For99] Kevin Ford. The number of solutions of $\phi(x) = m$. *Annals of Mathematics*, 150(1):283–311, 1999.

[Gau66] Carl Friedrich Gauss. *Disquisitiones Arithmeticae.* Yale University Press, New Haven, Conn.-London, 1966. Translated into English by Arthur A. Clarke, S. J.

[Gau76] Carl F. Gauss. *Werke.* Georg Olms Verlag, Hildesheim-New York, 1976. Ergänzungsreihe. IV. Briefwechsel C. F. Gauss–H. W. M. Olbers, II, Nachdruck der 1909 Auflage, herausgegeben von C. Schilling.

[Had93] Jacques Hadamard. Etude sur les propriétés des fonctions entières et en particulier d'une fonction considérée par riemann. *Jour. de Math. Pures et Appliquées*, pages 171–216, 1893.

[Had96] Jacques Hadamard. Sur la distribution des zéros de la fonction $\zeta(s)$ et ses conséquences arithmétiques. *Bulletin de la Société Mathématique de France*, 24:199–220, 1896.

[HB92] David R. Heath-Brown. Zero-free regions for Dirichlet L-functions, and the least prime in an arithmetic progression. *Proc. London Math. Soc. (3)*, 64(2):265–338, 1992.

[Hei34] Hans A. Heilbronn. On the class-number in imaginary quadratic fields. *The Quarterly Journal of Mathematics*, os-5(1):150–160, 01 1934.

[KJ47] Victor L. Klee Jr. On a conjecture of Carmichael. *Bull. Amer. Math. Soc.*, 53:1183–1186, 1947.

[Kou19] Dimitris Koukoulopoulos. *The Distribution of Prime Numbers*. Graduate Studies in Mathematics. American Mathematical Society, 2019.

[Leg09] Adrien-Marie Legendre. *Essai sur la théorie des nombres*. Cambridge Library Collection. Cambridge University Press, Cambridge, 2009. Reprint of the second (1808) edition.

[Lin44a] Yuri V. Linnik. On the least prime in an arithmetic progression. I. The basic theorem. *Rec. Math. [Mat. Sbornik] N.S.*, 15(57):139–178, 1944.

[Lin44b] Yuri V. Linnik. On the least prime in an arithmetic progression. II. The Deuring-Heilbronn phenomenon. *Rec. Math. [Mat. Sbornik] N.S.*, 15(57):347–368, 1944.

[LO77] Jeffrey C. Lagarias and Andrew M. Odlyzko. Effective versions of the Chebotarev density theorem. *Algebraic number fields: L-functions and Galois properties*, pages 409–464, 1977.

[LS06] Shanta Laishram and T. N. Shorey. The greatest prime divisor of a product of terms in an arithmetic progression. *Indag. Math. (N.S.)*, 17(3):425–436, 2006.

[May13] James Maynard. On the Brun-Titchmarsh theorem. *Acta Arith.*, 157(3):249–296, 2013.

[MV73] Hugh L. Montgomery and Robert C. Vaughan. The large sieve. *Mathematika*, 20:119–134, 1973.

[Pom74] Carl Pomerance. On Carmichael's conjecture. *Proc. Amer. Math. Soc.*, 43:297–298, 1974.

[SS58] Andrzej Schinzel and Waclaw Sierpiński. Sur certaines hypothèses concernant les nombres premiers. *Acta Arith.*, 4:185–208; erratum 5 (1958), 259, 1958.

[ST07] Tarlok N. Shorey and Rob Tijdeman. Prime factors of arithmetic progressions and binomial coefficients. In *Diophantine geometry*, volume 4 of *CRM Series*, pages 283–296. Ed. Norm., Pisa, 2007.

[SW94] Aaron Schlafly and Stan Wagon. Carmichael's conjecture on the Euler function is valid below $10^{10,000,000}$. *Math. Comp.*, 63(207):415–419, 1994.

[Syl12] James J. Sylvester. On arithmetical series. *Mathematical Papers*, 4:687–731, 1912.

[Tch52] Pafnuti L. Tchebichef. Mémoire sur les nombres premiers. *Journal de Mathématiques Pures et Appliquées*, pages 366–390, 1852.

[Tit30] Edward C. Titchmarsh. A divisor problem. *Rendiconti del Circolo Matematico di Palermo*, 54(1):414–429, 1930.

[Tru14] Tim Trudgian. Updating the error term in the prime number theorem. *The Ramanujan Journal*, 39:225–234, 2014.

[Tsc26] Nicolai G. Tschebotareff. Die Bestimmung der Dichtigkeit einer Menge von Primzahlen, welche zu einer gegebenen Substitutionsklasse gehören. *Math. Ann.*, 95(1):191–228, 1926.

[Wil88] John Williamson. *The Elements of Euclid: With Dissertations, Intended to Assist and Encourage a Critical Examination of These Elements, as the Most Effectual Means of Establishing a Juster Taste Upon Mathematical Subjects, Than that which at Present Prevails. Vol. II. By James Williamson, B.D.* T. Spilsbury, No. 57, Snowhill, 1788.

[Xyl11] Triantafyllos Xylouris. *Über die Nullstellen der Dirichletschen L-Funktionen und die kleinste Primzahl in einer arithmetischen Progression.* Universität Bonn, Mathematisches Institut, Bonn, 2011. Dissertation for the degree of Doctor of Mathematics and Natural Sciences at the University of Bonn, Bonn, 2011.

[Zam17] Asif A. Zaman. *Analytic estimates for the Chebotarev density theorem and their applications.* PhD thesis, University of Toronto, 2017.

Part IV

On The (M)iNTRU Assumption Over Finite Rings

The contents of this part have been published as:

Jim Barthel, Volker Müller and Răzvan Roşie (2021). On the MiNTRU assumption in the integer case. *ProvSec 2021*

Act IV: The obfuscation

Jay quickly carried out his second count and kept his promise to Missy by participating in her tea party. As usual, he had to vehemently deny her offer to eat some sand cake. Both had incredible fun pretending to be on the California beach slurping Mocktails; ma's creation was the best seller. Meanwhile, Jay's pa computed the total number of coins, albeit slightly distracted from the giggles and waves of laughter from Missy's room. Jay's notes revealed that the family collection contained 1478 pennies. Furthermore, a total of over 4000 coins and tokens were counted.

"Did you have fun counting the pennies?", asked Jay's pa at the dinner table. "Of course", replied Jay, "I can't wait to tell my friends about this." "Your friends?", injected pops, "you can't do that otherwise they will rob us!". "Oh Pieter!", exclaimed granny, "no one will steal your old rusty coins". "I still remember the twelfth of June 1963 when my lucky penny was stolen right after showing it to my buds in the club" responded pops angrily. "Not this story again", declared Jay's pa, "you lost it, that's all". "No, it was stolen. I know it!" reacted pops.

"You need to know Jay", said granny, "your pops is a bit paranoid. He won't share his treasures but prefers to hide them in the basement. He even encrypts the counting lists by taking another list, adding errors to this list, and finally multiplying it with the counting list." "You never know who's watching", explained pops, "and, by the way, if you try to make fun of me, do it correctly: I'm first multiplying the counting list with the other list and then I'm adding errors.". "It doesn't matter", sighed granny while shaking her head. Seemingly annoyed pops got up and went away. Right before leaving the room, he countered: "it does matter!".

Abstract IV

The inhomogeneous NTRU (iNTRU) assumption is a recent computational hardness assumption. Intuitively, it claims that first adding a random low norm error vector to a known gadget vector and then multiplying the result with a secret vector is sufficient to obfuscate the considered secret vector. The matrix inhomogeneous NTRU (MiNTRU) assumption replaces vectors with matrices and still claims to hide the secret. Albeit those assumptions strongly remind the well-known learning-with-errors (LWE) assumption, their hardness has not been studied in full detail yet. In this part, we break the basis case of the iNTRU and MiNTRU decision problems through an elementary q-ary lattice reduction attack. Concretely, we restrict the iNTRU assumption to finite integer rings and the MiNTRU assumption to vectors. This leads to a problem that we call (M)iNTRU. Starting from a challenge vector, we construct a particular q-ary lattice that shall reveal the nature of the challenge vector. Indeed, for a challenge vector following the uniform distribution, we obtain a random q-ary lattice for which it is unlikely that it contains an unusually short vector. For a challenge vector following the (M)iNTRU distribution, we obtain a special q-ary lattice that contains an unusually short vector. Thereby, elementary lattice reduction allows us to distinguish a random challenge vector from a synthetically constructed one. Subsequently, we describe how our attack can be generalized to the general iNTRU and MiNTRU problem and we highlight its inherent limitations. Ultimately, we conclude our development with a short comparison of the MiNTRU assumption with other well-known hardness assumptions and discuss some open questions.

Contents IV

Act IV **193**

Abstract IV **194**

Contents IV **195**

25 Introduction **197**

26 Inhomogeneous NTRU assumptions **199**

26.1 The inhomogeneous NTRU assumption 199

26.1.1 The iNTRU assumption 199

26.1.2 Applications 200

26.2 The matrix inhomogeneous NTRU assumption 202

26.2.1 The MiNTRU assumption 202

26.2.2 Applications 203

26.3 The (M)iNTRU problem 204

27 Attacking the (M)iNTRU assumption - First approach **205**

27.1 Lattice construction 205

27.2 A lattice basis 206

27.3 Case of a random challenge vector 207

27.4 Case of a (M)iNTRU challenge vector 208

27.4.1 First observation 208

27.4.2 Second observation 209

27.4.3 An unusually short lattice vector 209

27.5 Lattice reduction 210

27.5.1 Case of a random challenge vector 210

27.5.2 Case of a (M)iNTRU challenge vector 210

27.6 Conclusion 211

27.6.1 Comment on the error distribution 211

27.6.2 Comment on the success probability 211
27.7 Empirical tests 212

28 Attacking the (M)iNTRU assumption - Second approach 213
28.1 Lattice construction 213
28.2 Case of a random challenge vector 214
28.3 Case of a (M)iNTRU challenge vector 218
28.3.1 An observation 218
28.3.2 An unusually short lattice vector 218
28.4 Lattice reduction 219
28.4.1 Case of a random challenge vector 220
28.4.2 Case of a (M)iNTRU challenge vector 220
28.5 Conclusion 221
28.6 Empirical tests 221

29 Generalizations and limitations of our attacks 223
29.1 MiNTRU 223
29.1.1 Lattice construction 223
29.1.2 Case of a random challenge matrix 224
29.1.3 Case of a MiNTRU challenge matrix 225
29.1.4 Lattice reduction 226
29.1.5 Conclusion 228
29.1.6 Improvements and observations 228
29.2 Impact on cryptographic schemes 229

30 Comparisons with other hardness assumptions 231
30.1 NTRU 231
30.2 Learning-with-Errors 232
30.2.1 Multi-secret LWE 232
30.2.2 Hardness of multi-secret LWE 232
30.2.3 Connection to MiNTRU 233

31 Open Questions 235

Bibliography IV 237

Chapter 25

Introduction

The pillars of modern cryptography are objectively verifiable security notions. Whereas the definition of abstract procedures, such as *public-key cryptography* [DH76], is purposely designed in a universal manner, concrete instantiations, such as the *Diffie-Hellmann key exchange* [DH76], often require a link to presumably hard problems, such as the *discrete logarithm problem* [DH76]. This particular link is obtained through a *reduction* from the considered *problem* to a desired hardness *assumption*. Intuitively, the considered problem instance is insecure if the underlying assumption is wrong. For example, as long as discrete logarithms cannot be efficiently computed, the original Diffie-Hellmann key exchange is secure. This helps us to define secure parameters and to point out insecure cryptographic schemes. For illustration, *Shor's algorithm* [Sho94] shows the vulnerability of the Diffie-Hellmann key exchange towards quantum computers.

To not put all eggs in one basket, a myriad of security assumptions has been worked out. In particular, the quantum threat and the corresponding technological advances pushed the cryptographic community to find new hardness assumptions. Intriguing ideas such as *elliptic curve cryptography* [Mil85, Kob87] and *lattice-based cryptography* [Ajt96] have been conceptualized. Both of these cryptographic orientations play a major role in the U.S. national institute of standards and technology (NIST) post-quantum cryptography standarization [NIS17]. When focussing on lattice-based cryptography, we may distinguish two central hardness assumptions: the *Learning-with-Errors* (LWE) assumption [Reg05], and the *NTRU* assumption [HPS98]. Both of those assumptions are seemingly quantum secure and each presents its own advantages. Furthermore, both got equipped with a broad range of variants.

At AsiaCrypt 2019, *Genise et al* [GGH+19] increased the existing spectrum of post-quantum assumptions by two new computational hardness assumptions: the *inhomogeneous NTRU* (iNTRU) assumption and the *matrix inhomogeneous NTRU* (MiNTRU) assumption. Below, the intuition on these assumptions is given. The formal definitions can be found in Chapter 26.

Given a challenge vector $(a_0, \dots, a_\ell)$, the inhomogeneous NTRU decision problem (iNTRU) asks one to distinguish whether its entries were sampled uniformly at random from a chosen set of representatives $\mathcal{R}_q$ of a polynomial ring, or whether they were synthetically constructed. In the latter case, the challenge vector follows the iNTRU distribution that can be described as follows: first a secret invertible element $s \in \mathcal{R}_q^\times$ is chosen at random. Then, small norm error elements $e_i \in \mathcal{R}_q$ following a specific error distribution are sampled. Finally, the vector entries are defined by $a_0 := \left[s^{-1}e_0 \mod q_p\right]$ and $a_i := \left[s^{-1}(2^{i-1} - e_i) \mod q_p\right]$ for all $i \in \{1, \dots, \ell\}$ where the modulo operation first reduces the elements with respect to a polynomial $p(x)$ and then with respect to an integer q. The iNTRU assumption claims that both distributions are computationally indistinguishable.

The matrix inhomogeneous NTRU decision problem (MiNTRU) essentially replaces vectors over $\mathcal{R}_q$ by matrices with entries in $\mathbb{Z}_q := \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$. The MiNTRU assumption claims that deciding whether a matrix was chosen uniformly at random or stems from a particular sampling distribution is computationally infeasible. We remark that the iNTRU and MiNTRU problems intersect if vectors in $\mathbb{Z}_q^{\ell+1}$ are considered. We denote this elementary case by (M)iNTRU. As the existing security analysis of these problems is scarce, further research is required.

Hereinafter, we investigate the hardness of those two security assumptions in further detail. We succeed in developing two elementary lattice-based distinguishers for $\mathcal{R}_q = \mathbb{Z}_q$, breaking so the decisional hardness assumptions in their basis case. Our key idea consists in transforming the vector entries from a_i to $b_i := [2a_i - a_{i+1} \mod q] \equiv (-2e_i + e_{i+1})s^{-1} \mod q$, which makes the (M)iNTRU entries independent of the gadget terms 2^i. Then, we construct a particular q-ary lattice which contains an extremely short vector if the challenge vector follows the (M)iNTRU distribution. An observation on the shortest vector of q-ary lattices predicts that it is highly unlikely that a vector of this magnitude exists in a random q-ary lattice, which yields a natural distinction criterion. Remarkably, the size difference between the shortest vector of a random q-ary lattice and a q-ary lattice constructed from a (M)iNTRU vector is sufficiently large to be spotted through elementary lattice reduction. The upcoming development is based on the lattice notions defined in Chapter 6 and the particular results of Chapter 7.

Chapter 26

Inhomogeneous NTRU assumptions

Let us start by formally defining the problems of interest. Section 26.1 establishes the inhomogeneous NTRU assumptions and quickly revises their applications. Section 26.2 does the same for the matrix inhomogeneous NTRU assumptions. Section 26.3 introduces the problem of our study. We use the notations and conventions established in Chapter 1 and Chapter 2.

26.1 The inhomogeneous NTRU assumption

In this section, we (re-)define the inhomogeneous NTRU (iNTRU) assumption over polynomial rings, describe some variants, and outline its use.

26.1.1 The iNTRU assumption

The inhomogeneous NTRU problem has been introduced in [GGH+19, Section 4.1, formula (3)]. Its definition is based on a particular sampling process. Concretely, let $p(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0 \in \mathbb{Z}[x]$ be a monic polynomial. Let $q \in \mathbb{Z}_{\geq 2}$, $\ell = \lceil \log_2(q) \rceil$, and $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$. For all $g \in \mathbb{Z}[x]$, define $[g \mod q_p] \in \mathbb{Z}_q[x]$ by first reducing g modulo $p(x)$ to obtain a polynomial with degree strictly smaller than $\deg(p) = n$ and subsequently reducing the coefficients of the resulting polynomial modulo q to obtain coefficients in $\mathbb{Z}_q$. Let $\mathcal{R}_q = \{[g \mod q_p] \in \mathbb{Z}_q[x] \mid g \in \mathbb{Z}[x]\}$ and $\mathcal{R}_q^\times = \{g \in \mathcal{R}_q \mid \exists g^{-1} \in \mathcal{R}_q,\ [g^{-1}g \mod q_p] = 1\}$. Let χ be a symmetric distribution with support in $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ and standard deviation $\sigma_\chi = O(\sqrt{q})$. We refer to χ as the error distribution.

Definition 26.1 (iNTRU distribution)**.** Let s be sampled uniformly at random in $\mathcal{R}_q^\times$ and let $e_0, \ldots, e_\ell$ be sampled independently in $\mathcal{R}_q$ such that their coefficients follow the error distribution χ. Define the iNTRU distribution as the distribution of the vector $(a_0, \ldots, a_\ell)$ defined by

$$a_0 := [s^{-1}e_0 \mod q_p] \tag{26.1}$$

$$a_i := [s^{-1}(2^{i-1} - e_i) \mod q_p] \quad \forall i \in \{1, ..., \ell\}. \tag{26.2}$$

Given a vector $(a_0, \ldots, a_\ell)$ following the iNTRU distribution and the corresponding modulus q_p, the iNTRU search problem consists in finding the value of s. The iNTRU search assumption predicts that this can only be achieved with negligible probability. Given a vector $(x_0, \ldots, x_\ell)$ and a modulus q_p, the iNTRU decision problem consists in distinguishing whether the vector has been sampled following the iNTRU distribution or the uniform distribution over $\mathcal{R}_q^{\ell+1}$. The iNTRU decision assumption predicts that such a distinction can only be made with negligible probability. In [GGH$^+$19], the iNTRU decision problem is defined over abstract rings. However, for technical reasons, we restrict to polynomial rings. If $p(x) = x$, then, $\mathcal{R}_q = \mathbb{Z}_q$ and the underlying error distribution χ may be considered to be the discrete Gaussian distribution with standard deviation $\sigma_\chi = 2\sqrt{q}$. Furthermore, for practical reasons, one may consider shortened iNTRU vectors by removing the first vector entries.

26.1.2 Applications

The iNTRU assumptions have only been used once, namely in [GL20] where the pseudorandomness of two ring-based short integer solution lattice trapdoors is based on them. Below, we given an intuition on their construction. Indeed, we first illustrate the notion of trapdoors and describe the short integer solution problem. Subsequently, we portray the basic idea from [GL20] used to construct the short integer solution lattice trapdoors. Finally, we explain how the security of these trapdoors relates to the iNTRU assumptions.

Trapdoors

Intuitively, a *trapdoor* can be seen as an information that allows one to invert a particular function [DH76]. For example, let $\mathbf{A} \in (\mathcal{R}_q^{m \times m})_{inv}$ be an invertible matrix modulo q_p and let $f_\mathbf{A} : \mathcal{R}_q^m \to \mathcal{R}_q^m$ be defined by $f_\mathbf{A}(\mathbf{x}) := [\mathbf{A} \cdot \mathbf{x}^T \mod q_p]$, then $\mathbf{A}$ is a trapdoor for $f_\mathbf{A}$ as it allows us to recover $\mathbf{x}$ from $f_\mathbf{A}(\mathbf{x})$ by multiplying it on the left by its inverse.

Short Integer Solutions

The *Short Integer Solution* (SIS) problem (see [Ajt96, PR06, LS14]) is a cryptographic problem which, in the ring version, asks to find, for a given vector $\mathbf{a} \in \mathcal{R}_q^m$ and a bound value $\beta \in \mathbb{R}_{>0}$, a vector $\mathbf{x} \in \mathcal{R}_q^m$ such that $f_{\mathbf{a}}(\mathbf{x}) := \mathbf{a} \cdot \mathbf{x}^T \equiv 0 \mod q_p$ and $\|\mathbf{x}\|_\rho < \beta$ for a suitable metric $\| \cdot \|_\rho$.

Despite its difficulty, the problem can be tackled if some additional information on $\mathbf{a}$ is known. Assume that it is easy to solve the short integer solution problem for some particular function $f_{\mathbf{g}}$ where $\mathbf{g}$ is a known vector called the gadget. Assume further to know a low norm matrix $\mathbf{R}$, called a $\mathbf{g}$-trapdoor, such that $\mathbf{a} \cdot \mathbf{R} \equiv \mathbf{g} \mod q_p$. Then, the initial short integer solution problem can be easily solved. By our first assumption, a vector $\mathbf{x}$ such that $f_{\mathbf{g}}(\mathbf{x}) \equiv 0 \mod q_p$ and $\|\mathbf{x}\|_\rho < \beta$ can be efficiently determined and a transformed vector $\mathbf{x}' := [\mathbf{R} \cdot \mathbf{x} \mod q_p]$ can be computed using the known $\mathbf{g}$-trapdoor $\mathbf{R}$. Thus, it remains only to verify the size condition $\|\mathbf{x}'\|_\rho < \beta$, which, due to the low norm entries of $\mathbf{R}$, is usually satisfied. Often, the computation of $\mathbf{x}$ makes use of a discrete Gaussian sampling procedure such that multiple potential candidates can be generated increasing so the chance to find a suitable vector $\mathbf{x}'$. We refer to [GPV08] for a detailed description.

Their idea

[GL20] constructs two short integer solution trapdoors using the inherent trapdoor potential of the iNTRU distribution. Concretely, a shortened iNTRU vector $\mathbf{a} = (a_1, \dots, a_\ell)$ can be represented as $\mathbf{a} = [s^{-1}(\mathbf{g} + \mathbf{e}) \mod q_p]$ where $\mathbf{g} = (1, 2, 2^2, \dots, 2^{\ell-1})$ is the gadget vector, $\mathbf{e} = (e_1, \dots, e_\ell) \leftarrow_\chi \mathcal{R}_q^\ell$ is the error vector and $s \in \mathcal{R}_q^\times$ is the secret. Since $s\mathbf{a} \equiv \mathbf{g} + \mathbf{e} \mod q_p$ and we expect $\mathbf{g} + \mathbf{e} \approx \mathbf{g}$, the secret s is almost a $\mathbf{g}$-trapdoor for $\mathbf{a}$, falling short of $\mathbf{e}$.

Pseudorandomness

A trapdoor should be *pseudorandom*, which means that it should be hard to be guessed. Translated to the construction above, this means that s should not be deducible from $\mathbf{a} = [s^{-1}(\mathbf{g} + \mathbf{e}) \mod q_p]$ and, in particular, $\mathbf{a}$ should be indistinguishable from a uniformly at random sampled vector. These properties follow directly from the iNTRU search and decision assumption. By devising two distinct preimage sampling processes for $\mathbf{s}$, [GL20] obtains two distinct trapdoor schemes. We note that, for technical reasons, the final trapdoor constructions replace the gadget $\mathbf{g}$ by an approximate gadget $\mathbf{f} = (2^j, \dots, 2^{\ell-1})$ for some $j \in \mathbb{N}_{>1}$, but the key idea remains the same. We refer to [GL20, Section 3] for the detail of their constructions.

26.2 The matrix inhomogeneous NTRU assumption

In this section, we (re)define the matrix inhomogeneous NTRU (MiNTRU) assumption, describe some variants, and outline its use.

26.2.1 The MiNTRU assumption

The matrix inhomogeneous NTRU problem was introduced in [GGH+19, Section 4.1, formula (4)]. Its definition is based on a particular sampling process. Concretely, let $q \in \mathbb{Z}_{\geq 2}$, $n \in \mathbb{Z}_{\geq 1}$, $\ell = \lceil \log_2(q) \rceil$, $m = n(\ell + 1)$, and $\mathbf{G} = [\mathbf{0}|\mathbf{I}|2\mathbf{I}|...|2^{\ell-1}\mathbf{I}] \in \mathbb{Z}^{n \times m}$ be a particular matrix called the gadget matrix. Let χ be a symmetric distribution with support in $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ and standard deviation $\sigma_\chi = O(\sqrt{q})$. We refer to χ as the error distribution. Let $(\mathbb{Z}_q)^{n \times n}_{inv}$ denote the set of matrices in $\mathbb{Z}_q^{n \times n}$ that are invertible modulo q.

Definition 26.2 (MiNTRU distribution)**.** Let $\mathbf{S}$ be sampled uniformly at random in $(\mathbb{Z}_q)^{n \times n}_{inv}$ and let $\mathbf{E}$ be an $n \times m$ matrix whose entries have been sampled independently following the error distribution χ. Set the MiNTRU distribution as the distribution of the matrix $\mathbf{A}$ defined by

$$\mathbf{A} := \left[\mathbf{S}^{-1} \times (\mathbf{G} - \mathbf{E}) \mod q\right] \tag{26.3}$$

where the modulo operation returns for each matrix entry the unique representative in $\mathbb{Z}_q$.

The MiNTRU search problem consists in retrieving the hidden secret matrix $\mathbf{S}$ from a MiNTRU matrix $\mathbf{A}$ and its modulus q, and the decision problem asks one to distinguish a uniformly at random sampled matrix $\mathbf{X}$ from a MiNTRU matrix $\mathbf{A}$. The MiNTRU assumptions claim that these problems can only be solved with negligible probability. In [GGH+19] only the decision problem was defined, but the search variant may be of cryptographic use. Following a suggestion of [GGH+19] the underlying error distribution χ may be considered to be the discrete Gaussian distribution with standard deviation $\sigma_\chi = 2\sqrt{q}$. Although no standard parameters have been defined, the original article suggests to use the matrix dimension $n = O(q^{1/4})$. Variants of the MiNTRU problem may be obtained by removing some parts of the MiNTRU matrix. For example, the *small secret* variant is obtained by removing the first $n \times n$ block of the MiNTRU matrix and sampling the invertible secret matrix $\mathbf{S}$ such that its entries follow the error distribution χ.

26.2.2 Applications

The MiNTRU assumptions have only been invoked in their original formalization paper [GGH+19] where they are used to prove the semantic security of a new homomorphic encryption scheme for finite automata. Below, we illustrate their construction. More precisely, we first give an intuition on homomorphic encryption schemes for non-deterministic finite automata. Then, we describe the basic encryption procedure of [GGH+19] and its corresponding homomorphic evaluation. At last, we explain how the security of the basic encryption procedure relates to the MiNTRU assumptions.

Homomorphic encryption and non-deterministic finite automata

Homomorphic encryption (HE) [RAD78] enables computations over encrypted data leaving the result under encrypted form. A particular case is given for the evaluation of encrypted *non-deterministic finite automata* (NFA) [RS59] that can be interpreted as a matrix product. Intuitively, a homomorphic encryption scheme for non-deterministic finite automata allows one to evaluate the product $\left[\left(\prod_{i=1}^{k} \mathbf{M}_{k+1-i}\right) \cdot \mathbf{v} \mod q\right]$. However, instead of evaluating this product in plain, the computation is carried out over encryptions of the matrices $\mathbf{M}_i \in \mathbb{Z}_q^{n\times n}$ and the vector $\mathbf{v} \in \mathbb{Z}_q^n$. [GGH+19] devises a suitable scheme for such a computation.

Their basic encryption procedure

The encryption of a given message $\mathbf{M} \in \mathbb{Z}_q^{n\times n}$ takes place in two steps: first, a secret matrix $\mathbf{S} \in (\mathbb{Z}_q)_{inv}^{n\times n}$ is chosen uniformly at random and an error matrix $\mathbf{E} \in \mathbb{Z}_q^{n\times m}$ with $m = n \times (\ell + 1)$ is sampled with respect to the error distribution χ. Then, a gadget matrix $\mathbf{G} = [\mathbf{0}|\mathbf{I}|2\mathbf{I}|...|2^{\ell-1}\mathbf{I}] \in \mathbb{Z}^{n\times m}$ is constructed and the ciphertext $\mathbf{C} := \left[\mathbf{S}^{-1} \times (\mathbf{M} \times \mathbf{G} + \mathbf{E}) \mod q\right]$ is computed. The decryption of a ciphertext $\mathbf{C}$ is obtained by computing $[\mathbf{S} \times \mathbf{C} - \mathbf{E} \mod q] \equiv \mathbf{M} \times \mathbf{G} \mod q$ and recovering $\mathbf{M}$ through a known trapdoor of the gadget matrix $\mathbf{G}$. Similarly, a vector $\mathbf{v} \in \mathbb{Z}_q^n$ can be encrypted by setting $\mathbf{c} := \left[\mathbf{S}^{-1} \times (\mathbf{v} + \mathbf{e}) \mod q\right]$ where $\mathbf{e} \in \mathbb{Z}_q^n$ is sampled with respect to the error distribution χ.

Chained encryption for homomorphic evaluation

Homomorphic evaluation of an encrypted product $\left(\prod_{i=1}^{k} \mathbf{M}_{k+1-i}\right) \cdot \mathbf{v}$, is achieved through a recursive process. First $k + 1$ secret keys $(\mathbf{S}_i, \mathbf{E}_i)$ with

$i \in \{0, \ldots, k\}$ are chosen uniformly at random. Then, the initial vector $\mathbf{v}$ is encrypted as $\mathbf{c} := \left[\mathbf{S}_0^{-1} \times (\beta\mathbf{v} + \mathbf{e}) \mod q\right]$. Next, the matrices $\mathbf{M}_i$ are iteratively encrypted by $\mathbf{C}_i := \left[\mathbf{S}_i^{-1} \times (\mathbf{M}_i \times \mathbf{S}_{i-1} \times \mathbf{G} + \mathbf{E}_i) \mod q\right]$ for all $i \in \{1, \ldots, k\}$. The homomorphic evaluation can now take place starting from $\mathbf{c}_0 := \mathbf{c}$ and recursively computing $\mathbf{c}_i := \left[\mathbf{C}_i \times \mathbf{G}^{-1}(\mathbf{c}_{i-1}) \mod q\right]$ for all $i \in \{1, \ldots, k\}$ where $\mathbf{G}^{-1}(c_i)$ denotes a discrete Gaussian vector, which satisfies $\mathbf{G} \times \mathbf{G}^{-1}(\mathbf{c}_i) \equiv \mathbf{c}_i \mod q$. The decryption of the final ciphertext $\mathbf{c}_k$ corresponds to the desired output $\left[\left(\prod_{i=1}^{k} \mathbf{M}_{k+1-i}\right) \cdot \mathbf{v} \mod q\right]$.

Security of the new scheme

Semantic security of the underlying encryption procedure implies semantic security of the homomorphic encryption scheme. The basic encryption procedure above is linked to the MiNTRU assumptions as an encrypted matrix $\mathbf{C} = \left[\mathbf{S}^{-1} \times (\mathbf{M} \times \mathbf{G} + \mathbf{E}) \mod q\right]$ consists almost in a MiNTRU matrix $\mathbf{A} = \left[\mathbf{S}^{-1} \times (\mathbf{G} + \mathbf{E}) \mod q\right]$, except for the additional factor $\mathbf{M}$. Through a convolution of error distributions, the security of the encryption scheme can be reduced to the MiNTRU assumptions [GGH+19, Proposition 4.2.].

26.3 The (M)iNTRU problem

Choosing in Section 26.1.1 the polynomial $p(x) = x$ and setting in Section 26.2.1 the matrix dimension to $n = 1$ leads to the same problem. Concretely, let $q \in \mathbb{Z}_{\geq 2}$, $\ell = \lceil \log_2(q) \rceil$ and let χ be a symmetric distribution with support in $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ and standard deviation $\sigma_\chi = O(\sqrt{q})$. We refer to χ as the error distribution.

Definition 26.3 ((M)iNTRU distribution)**.** Let s be sampled uniformly at random in $\mathbb{Z}_q^\times$ and let $e_0, \ldots, e_\ell$ be sampled independently in $\mathbb{Z}_q$ following the error distribution χ. Set the (M)iNTRU distribution as the distribution of the vector $(a_0, \ldots, a_\ell)$ defined by

$$a_0 := \left[s^{-1} e_0 \mod q\right], \tag{26.4}$$

$$a_i := \left[s^{-1}(2^{i-1} - e_i) \mod q\right] \quad \forall i \in \{1, ..., \ell\}. \tag{26.5}$$

Similar to the iNTRU and MiNTRU search problems, the (M)iNTRU search problem consists in retrieving the hidden secret value s from a (M)iNTRU vector $(a_0, \ldots, a_\ell)$ and its modulus q, and the decision problem only asks to distinguish a random vector $(x_0, \ldots, x_\ell)$ from a (M)iNTRU vector $(a_0, \ldots, a_\ell)$. In the upcoming section, we focus on the (M)iNTRU decision problem and we show that it can be solved through an elementary lattice attack.

Chapter 27

Attacking the (M)iNTRU assumption - First approach

In this chapter, we develop a lattice attack against the (M)iNTRU decision assumption. The attack constructs a particular q-ary lattice and employs elementary lattice reduction to approximate its shortest vector. The size of this approximation helps us to distinguish (M)iNTRU vectors from random ones. The key for this distinction is Theorem 7.5 which gives a probabilistic estimate on the expected size of the shortest vector. Furthermore, in case a (M)iNTRU vector has been detected, it often allows us to filter out the underlying secret s. Although our development is based on full-length challenge vectors, it can be modified to apply to shortened vectors as well.

27.1 Lattice construction

Let $\mathbf{x} = (x_0, \ldots, x_\ell)$ denote a challenge vector whose entries either follow the uniform distribution or the (M)iNTRU distribution over $\mathbb{Z}_q^{\ell+1}$ where $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$. We construct a new vector $\mathbf{y} := (y_0, \ldots, y_{\ell-1})$ by setting

$$y_0 := [x_0 \mod q] \quad \text{and} \quad y_i := [2x_i - x_{i+1} \mod q]. \tag{27.1}$$

for all $i \in \{1, \ldots, \ell - 1\}$ where the modulo operation returns the unique representative of the result in $\mathbb{Z}_q$. Subsequently, using the notation from Section 7.1, we construct the $\ell \times \ell$ q-ary row lattice:

$$\Lambda_q(\mathbf{y}) = \left\{ \mathbf{t} \in \mathbb{Z}^\ell \;\middle|\; \mathbf{t} \equiv t\mathbf{y} \mod q \text{ for some } t \in \mathbb{Z} \right\}. \tag{27.2}$$

27.2 A lattice basis

Next, we wish to find an explicit lattice basis of $\Lambda_q(\mathbf{y})$. To obtain this basis, we require one entry of $\mathbf{y}$ to be invertible modulo q. In case no entry of $\mathbf{y}$ is invertible modulo q, the upcoming transformation cannot be carried out and the subsequent analysis does not hold. Although the analysis could still be adapted to that situation, we suggest using in this case our second attack described in Chapter 28. However, as the following proposition shows, it is rather unlikely to not find an invertible element of $\mathbf{y}$ modulo q.

Proposition 27.1. *The probability that a uniformly at random sampled vector $(y_0, \ldots, y_{\ell-1}) \in \mathbb{Z}_q^\ell$ contains at least one invertible element modulo q tends to 1 for increasing q.*

Proof. The probability that a random $y \in \mathbb{Z}_q$ is invertible modulo q is $\frac{\varphi(q)}{q}$ where φ denotes the *Euler totient function* [Gau66, §38]. Thus, the probability that none of the entries in $(y_0, \ldots, y_{\ell-1})$ is invertible is only $\left(1 - \frac{\varphi(q)}{q}\right)^\ell$. As by [RS62],

$$\varphi(q) > \frac{q}{e^\gamma \log(\log(q)) + \frac{3}{\log(\log(q))}}, \tag{27.3}$$

the probability is upper bounded by

$$\left(1 - \frac{1}{e^\gamma \log(\log(q)) + \frac{3}{\log(\log(q))}}\right)^\ell \tag{27.4}$$

where $\gamma \approx 0.57721566$ is *Eulers constant* [Lag13]. An elementary function analysis shows that this upper bound tends to 0 for increasing q. □

Remark 27.2. *We note that for most values of q, the probability that none of the entries of $\mathbf{y}$ is invertible is far smaller than the upper bound outlined in Equation (27.4). For example, if q is prime, the probability is only $\left(\frac{1}{q}\right)^\ell$.*

Thus, assume that $t \in \{0, \ldots, \ell-1\}$ is an index such that y_t is invertible modulo q, or, in other words, such that $\gcd(y_t, q) = 1$. Then, we transform the vector $\mathbf{y}$ to obtain a new vector $\mathbf{z} := (z_0, \ldots, z_{\ell-1})$ by setting

$$z_i := \left[y_t^{-1} y_i \mod q\right] \quad \forall i \in \{0, \ldots, \ell-1\} \tag{27.5}$$

where $z_t = 1$. With this new vector, we can develop an explicit lattice basis of $\Lambda_q(\mathbf{y})$.

Lemma 27.3. *With* $\mathbf{y}$ *and* $\mathbf{z}$ *as above, we have*

$$\Lambda_q(\mathbf{y}) = \mathcal{L}\begin{pmatrix} z_0 & \dots & z_{t-1} & 1 & z_{t+1} & \dots & z_{\ell-1} \\ q & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & q & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & q & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & q \end{pmatrix}.$$

Proof. To prove that both lattices coincide, it suffices to show that each generating vector of one lattice is also contained inside the other one. By definition, $\Lambda_q(\mathbf{y})$ is generated by the canonical vectors $(q, 0, \dots, 0), \dots, (0, \dots, 0, q)$ and $\mathbf{y}$. The lattice on the right hand side is generated by the same vectors, but replaces $\mathbf{y}$ and the canonical vector with q on its $(t+1)$-th entry with the vector $\mathbf{z}$.

First, we note that $\mathbf{z} \in \Lambda_q(\mathbf{y})$. Indeed, multiplying $\mathbf{y}$ by y_t^{-1} and subtracting as often as needed the canonical vectors $(q, 0, \dots, 0), \dots, (0, \dots, 0, q)$ from the result shows that $\mathbf{z} \in \Lambda_q(\mathbf{y})$. Therefore, the lattice on the right is a sublattice of $\Lambda_q(\mathbf{y})$.

Reciprocally, we observe that the canonical vector with q on its $(t+1)$-th entry is obtained by subtracting for each $i \in \{0, \dots, t-1, t+1, \dots, \ell\}$ the canonical vector with q on its i-th entry exactly z_i times from $q\mathbf{z}$. Furthermore, multiplying $\mathbf{z}$ by y_t and subtracting the canonical vectors from the result as often as necessary shows that $\mathbf{y}$ is contained in the lattice on the right, showing that $\Lambda_q(\mathbf{y})$ is a sublattice of the lattice on the right. □

In the two subsequent sections, we study the differences of $\Lambda_q(\mathbf{y})$ assuming that the initial challenge vector $\mathbf{x}$ either follows the uniform or the (M)iNTRU distribution. During this analysis, we freely switch between the two expressions of $\Lambda_q(\mathbf{y})$ to enjoy the properties of a q-ary lattice, but also the properties that can be deduced from its particular basis given in Lemma 27.3.

27.3 Case of a random challenge vector

Assume that our initial challenge vector $\mathbf{x} = (x_0, \dots, x_\ell)$ was sampled uniformly at random. Then, the entries of our constructed vector $\mathbf{y} \in \mathbb{Z}_q^{1\times\ell}$ still follow the uniform distribution. By Theorem 7.5, we have

$$\mathbb{P}\left(\lambda_1(\Lambda_q(\mathbf{y})) \geq \min\left\{q, \sqrt{\frac{\ell}{8\pi e}q^{\frac{\ell-1}{\ell}}}\right\}\right) \geq 1 - \frac{1}{\sqrt{\pi\ell}}2^{-\ell}. \quad (27.6)$$

where $\lambda_1(\Lambda_q(\mathbf{y}))$ denotes the first lattice minimum of $\Lambda_q(\mathbf{y})$ in the Euclidean norm. Thereby, for sufficiently large ℓ, the shortest lattice vector can be expected to satisfy

$$\lambda_1(\Lambda_q(\mathbf{y})) \geq \min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}. \tag{27.7}$$

27.4 Case of a (M)iNTRU challenge vector

Assume next that the challenge vector $\mathbf{x} = (x_0, \ldots, x_\ell)$ was synthetically constructed following the $\mathsf{(M)iNTRU}$ distribution. We will show that in this case, $\Lambda_q(\mathbf{y})$ contains an unusually short lattice vector that is considerably smaller than the expected size of its random equivalent.

27.4.1 First observation

We recall that a synthetically constructed vector $\mathbf{a} := (a_0, \ldots, a_\ell) \in \mathbb{Z}_q^{\ell+1}$ following the $\mathsf{(M)iNTRU}$ distribution satisfies

$$a_0 := \left[s^{-1}e_0 \mod q\right] \quad \text{and} \quad a_i := \left[s^{-1}(2^{i-1} - e_i) \mod q\right] \tag{27.8}$$

for all $i \in \{1, \ldots, \ell\}$ where $e_0, \ldots, e_\ell$ denote random errors sampled from the symmetric error distribution χ producing with overwhelming probability elements that are small in absolute value. Thereby, the transformation

$$y_0 := \left[s^{-1}e_0 \mod q\right] \quad \text{and} \quad y_i := \left[s^{-1}(-2e_i + e_{i+1}) \mod q\right] \tag{27.9}$$

for all $i \in \{1, \ldots, \ell - 1\}$ produces elements where the numerators e_0 and $-2e_i + e_{i+1}$ for all $i \in \{1, \ldots, \ell - 1\}$ are still quite small compared to q. To be precise, the numerators follow the distribution χ' where:

1. The mean $\mu_{\chi'}$ of χ' is

$$\mu_{\chi'} = -2\mu_\chi + \mu_\chi = 0$$

   where the first equality follows from the sum of random variables and since a_i and a_{i+1} follow the distribution χ, and the second equality comes from the symmetry of χ implying that $\mu_\chi = 0$.

2. The variance $\sigma^2_{\chi'}$ of χ' is given by

$$\sigma^2_{\chi'} = 3\sigma^2_\chi$$

   since a_i and a_{i+1} follow the same distribution χ.

As χ produces with overwhelming probability elements with a small absolute value, so does χ'. Thus, we can expect the numerators to be quite small when compared to the modulus q.

27.4.2 Second observation

Considering the transformation leading to $\mathbf{z}$, we observe that its entries are given by

$$z_i = \left[(-2e_i + e_{i+1})e_t'^{-1} \mod q\right] \quad \forall i \in \{0, \ldots, \ell-1\} \tag{27.10}$$

where $e_t' = e_0$ if $t = 0$ and $e_t' = -2e_t + e_{t+1}$ if $t \in \{1, \ldots, \ell-1\}$. Thus, each entry z_i can be expressed as the quotient of two small norm error elements.

27.4.3 An unusually short lattice vector

Our two observations show that our lattice contains the unusually short vector

$$\mathbf{v} := (e_0, (-2e_1 + e_2), \ldots, (-2e_{\ell-1} + e_\ell)) \tag{27.11}$$

that can be obtained from $e_t'\mathbf{z}$ by subtracting the canonical vectors $(q, 0, \ldots, 0), \ldots, (0, \ldots, 0, q)$ as often as needed. If the error entries are bounded in absolute value by some constant $K > 0$, for example

$$\max\{|e_0|, |-2e_1 + e_2|, \ldots, |-2e_{\ell-1} + e_\ell|\} \leq K, \tag{27.12}$$

then the size of v is upper bounded by

$$\|\mathbf{v}\|_2 \leq \sqrt{\ell K^2} \leq \sqrt{\ell}\, K. \tag{27.13}$$

This allows us to give explicit bounds for the error values such that the size of $\mathbf{v}$ is smaller than the expected value in Equation (27.7).

Proposition 27.4. *If* $\max\{|e_0|, |-2e_1+e_2|, \ldots, |-2e_{\ell-1}+e_\ell|\} \leq \min\left\{\frac{q}{\sqrt{\ell}}, \frac{1}{\sqrt{8\pi e}} q^{(\ell-1)/\ell}\right\}$, *then the target vector* $\mathbf{v}$ *is shorter than* $\min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}} q^{\frac{\ell-1}{\ell}}\right\}$.

Proof. Replace K in Equation (27.13) with $\min\left\{\frac{q}{\sqrt{\ell}}, \frac{1}{\sqrt{8\pi e}} q^{(\ell-1)/\ell}\right\}$. □

In practice, the error terms are chosen to be of size $O(\sqrt{q})$, which implies that the required size condition is almost always satisfied.

27.5 Lattice reduction

As the preceding development yields good estimates on the expected size of the shortest lattice vector of $\Lambda_q(\mathbf{y})$, we can investigate the effect of elementary lattice reduction on $\Lambda_q(\mathbf{y})$.

27.5.1 Case of a random challenge vector

If $\mathbf{x}$ was sampled uniformly at random, then we concluded in Equation (27.7) that the shortest lattice vector of $\Lambda_q(\mathbf{y})$ can be expected to be of size $\lambda_1(\Lambda_q(\mathbf{y})) \geq \min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$. Therefore, the shortest LLL reduced basis vector is at least of the same size.

27.5.2 Case of a (M)iNTRU challenge vector

Slightly decreasing the error bound K in Equation (27.13) guarantees that the first LLL reduced vector is smaller than the expected random value. Indeed, generally, the first LLL reduced vector with factor δ does not correspond to the smallest lattice vector but only consists in a good approximation of it. More precisely, the first LLL reduced vector $\mathbf{w}_1$ of $\Lambda_q(\mathbf{y})$ satisfies $\|\mathbf{w}_1\| \leq \alpha^{\frac{\ell-1}{2}}\lambda_1(\Lambda_q(\mathbf{y}))$ where $\alpha = \frac{1}{\delta - \frac{1}{4}}$. However, in practice, this artificial blow-up is barely observed. α can be decreased by increasing δ, but such an improvement is limited to $\delta < 1$. For the sake of explicit results, we consider hereinafter $\delta = \frac{63}{64} < 0.99$ resulting in $\alpha = \frac{64}{47} < \sqrt{2}$. With these parameters, we deduce from Equation (27.13) that

$$\|\mathbf{w}_1\|_2 \leq \alpha^{\frac{\ell-1}{2}}\sqrt{\ell}K \leq 2^{\frac{\ell-1}{4}}\sqrt{\ell}K \leq 2^{\frac{\log_2(q)}{4}}\sqrt{\ell}K \leq q^{\frac{1}{4}}\sqrt{\ell}K. \tag{27.14}$$

Adapting K accordingly leads to the following conclusion.

Proposition 27.5. *If* $\max\{|e_0|, |-2e_1+e_2|, \ldots, |-2e_{\ell-1}+e_\ell|\} \leq \min\left\{\frac{q^{3/4}}{\sqrt{\ell}},\right.$ $\left.\frac{1}{\sqrt{8\pi e}}q^{\frac{3\ell-4}{4\ell}}\right\}$*, then* $\|\mathbf{w}_1\|_2$ *is smaller than* $\min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$.

Proof. Replace K in Equation (27.14) by $\min\left\{\frac{q^{3/4}}{\sqrt{\ell}}, \frac{1}{\sqrt{8\pi e}}q^{\frac{3\ell-4}{4\ell}}\right\}$. □

We note again that [GGH+19] suggested $K = O(\sqrt{q})$ so that we can expect the condition of Proposition 27.5 to hold in practice.

27.6 Conclusion

We conclude that:

1. If the challenge vector was sampled uniformly at random, the first LLL reduced vector can be expected with high probability (see Equation (27.6)) to be lower bounded by $\min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$.
2. If the challenge vector follows the (M)iNTRU distribution with a sufficiently small error bound, the first LLL reduced vector is deterministically smaller than $\min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$ (see Proposition 27.5).

Hence, conditioned on the symmetric error distribution χ, we can distinguish between a challenge vector that was sampled uniformly at random and a challenge vector that follows the (M)iNTRU distribution by comparing the length of the first LLL reduced vector with $\min\left\{q,\ \sqrt{\frac{\ell}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$.

27.6.1 Comment on the error distribution

We point out that our development did not use a particular error distribution. The behaviour of χ not being known makes the analysis harder but still allows for our conclusion. In practical applications of the (M)iNTRU assumption, the error distribution is known as it needs to be hard-coded into the resulting primitives. This additional information may be used to obtain tighter bounds and impossibility results. For example, most implementations of the Discrete Gaussian distribution set the probability for elements sampled outside a given error range to 0 such that K in *Equation* (27.13) can be made precise.

27.6.2 Comment on the success probability

If the maximal error term is sufficiently small, then we are guaranteed that our distinguisher always recognizes (M)iNTRU challenge vectors. Indeed, if

$$\max\{|e_0|, |e_1|, \ldots, |e_\ell|\} \leq \min\left\{\frac{q^{3/4}}{3\sqrt{\ell}}, \frac{1}{3\sqrt{8\pi e}}q^{\frac{3\ell-4}{4\ell}}\right\}, \tag{27.15}$$

then the size condition of Proposition 27.5 holds and (M)iNTRU challenge vectors are correctly assessed. Thus, in this case the overall success probability depends on wrongly assessed random challenge vectors only. The probability of such bad evaluations is given in Equation (27.6). Assuming that it is decided uniformly at random whether the challenge vector stems from the uniform distribution or the (M)iNTRU distribution, the overall success probability is higher than $1 - \frac{1}{\sqrt{\pi\ell}} 2^{-(\ell+1)}$.

If the maximal error term is larger than the upper bound in Equation (27.15), then only a heuristic success probability can be obtained. The probability of success in correctly assessing random challenge vectors does not change, but correct identification of (M)iNTRU challenge vectors may be at risk. As the size of a single error element is not important, but only the size of the resulting error vector $\mathbf{v}$ matters, there is a high probability that an isolated large vector entry does not imply a violation of the vector size condition for $\mathbf{v}$. Furthermore, our development uses the worst-case approximation of LLL reduced vectors, which is not observed in practice. Thus, if the error distribution range or its variation is sufficiently small, then a high success rate can be expected.

27.7 Empirical tests

An implementation of this distinguisher in SAGEMATH (distinguisher1) and the corresponding statistics can be found at:

https://orbilu.uni.lu/handle/10993/47990

Practical experiments confirm the above conclusions but show another surprising side effect, namely that almost always the secret s can be obtained from (M)iNTRU challenge vectors. Indeed, typically, the first LLL reduced vector $\mathbf{w}_1$ corresponds to either $\mathbf{v}$ or its negative. If so, choosing the error term in the t-th position e'_t, and computing $\left[e'_t y_t^{-1} \mod q\right] = \pm s$ reveals the secret s. Thus, the corresponding (M)iNTRU search problem seems to be solvable with the same lattice method. However, a theoretical confirmation of this observation remains open.

Chapter 28

Attacking the (M)iNTRU assumption - Second approach

In this chapter, we describe a second attack against the (M)iNTRU assumption. It may be applied if the attack described in Chapter 27 cannot be used or if the result of this attack needs to be double-checked. The upcoming attack is still based on the same idea as the one described in Chapter 27 but avoids the invertibility condition by constructing another lattice. The tradeoff for the universality of this second attack is its dependence on heuristics.

28.1 Lattice construction

Let $\mathbf{x} = (x_0, \ldots, x_\ell)$ denote again a (M)iNTRU challenge vector following either the uniform distribution or the (M)iNTRU distribution over $\mathbb{Z}_q^{\ell+1}$ where $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$. Similar than in our first attack, we construct the vector $\mathbf{y} = (y_0, \ldots, y_{\ell-1})$ by setting

$$y_0 := [x_0 \mod q] \quad \text{and} \quad y_i := [2x_i - x_{i+1} \mod q], \tag{28.1}$$

for all $i \in \{1, \ldots, \ell - 1\}$ where the modulo operation returns the unique representative of the result in $\mathbb{Z}_q$. Contrary to our first attack, we construct now the $(\ell+1)\times(\ell+1)$ row lattice:

$$\Lambda = \mathcal{L}\begin{pmatrix} y_0q & y_1q & y_2q & \dots & y_{\ell-1}q & 1 \\ q^2 & 0 & 0 & \dots & 0 & 0 \\ 0 & q^2 & 0 & \dots & 0 & 0 \\ 0 & 0 & q^2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & q^2 & 0 \end{pmatrix} \tag{28.2}$$

Clearly, Λ is a q^2-ary lattice as the vector $(0,\dots,0,q^2)$ can be obtained by multiplying the first row by q^2 and subtracting the other rows as often as needed.

28.2 Case of a random challenge vector

Assume that our initial challenge vector $\mathbf{x} = (x_0, ..., x_\ell)$ was sampled uniformly at random. Then, the entries of $\mathbf{y}$ follow the uniform distribution. Thus, the first row of our lattice basis could be thought of as behaving, up to the common factor q and the last entry, randomly. This would motivate the use of Theorem 7.5 to conclude that with high probability

$$\lambda_1(\Lambda) \geq \min\left\{q^2,\ \sqrt{\frac{\ell+1}{8\pi e}}q^{\frac{2\ell}{\ell+1}}\right\}. \tag{28.3}$$

However, considering the special shape of the first row of the basis matrix of Λ, the lattice cannot be seen as random. This precaution is justified since Λ contains an unusually short vector, namely $\mathbf{v}_1 := (0,\dots,0,\frac{q}{g})$ where $g = \gcd(y_0,\dots,y_{\ell-1},q)$, which is obtained by multiplying the first row by $\frac{q}{g}$ and subtracting the other rows as often as needed. Thus, $\lambda_1(\Lambda) \leq \frac{q}{g} \leq q$ directly contradicting the desired conclusion.

Nonetheless, we will see that the lower bound in Equation (28.3) is a good estimate for the subsequent successive minima and in particular for $\lambda_2(\Lambda)$. Concretely, we will prove that, apart from the trivially short vector $\mathbf{v}_1 = (0,\dots,0,\frac{q}{g})$ and its multiples, it is unlikely to find another vector smaller than the expected value in Equation (28.3).

Lemma 28.1. *Let $B \leq \frac{q}{2}$ be an integer and $S \subseteq \mathbb{Z}$ be fixed. Choose $r \in \mathbb{Z}_{q^2} = \mathbb{Z}\cap\left(-\frac{q^2}{2},\frac{q^2}{2}\right]$ uniformly at random and for each $i \in \{0,\dots\ell-1\}$, let $y_i \in \mathbb{Z}_q = \mathbb{Z}\cap\left(-\frac{q}{2},\frac{q}{2}\right]$ be chosen uniformly at random. Set*

$$\mathbf{y} = (y_0q, y_1q, \dots, y_{\ell-1}q, 1).$$

Then, the probability that $r \in S$ and the Euclidean norm of the vector $\left[r\mathbf{y} \mod q^2\right]$ is at most Bq, where the modulo reduction returns for each entry the unique representative in $\mathbb{Z}_{q^2}$, is upper bounded by

$$\sum_{\substack{\beta_\ell=-Bq \\ \beta_\ell \in S}}^{Bq} \frac{\ell(2\lfloor B/\gcd(\beta_\ell, q)\rfloor + 1)}{q^2} \left(\frac{\gcd(\beta_\ell, q)}{q}\right)^{\ell}.$$

Proof. We observe that by switching from the Euclidean norm to the infinity norm, we have

$$\begin{aligned} &\mathbb{P}\left(\left(\left\|\left[r\mathbf{y} \mod q^2\right]\right\|_2 \leq Bq\right) \wedge (r \in S)\right) && (28.4)\\ &\leq \mathbb{P}\left(\left(\max\left\{\left|\left[r\mathbf{y} \mod q^2\right]\right|\right\} \leq Bq\right) \wedge (r \in S)\right) && (28.5)\end{aligned}$$

where the maximum is taken over all modulo q^2 reduced entries of $r\mathbf{y}$. Equation (28.5) is equal to the following probability of intersection of events

$$\mathbb{P}\left(\left(\bigwedge_{i=0}^{\ell-1} \underbrace{\left(\left|\left[ry_iq \mod q^2\right]\right| \leq Bq\right)}_{=:C_i}\right) \wedge \underbrace{\left(\left|\left[r \mod q^2\right]\right| \leq Bq\right)}_{=:C_\ell} \wedge (r \in S)\right). \quad (28.6)$$

Each event $C_0, \ldots, C_{\ell-1}$ in this probability statement can be written as a union of events:

$$C_i = \bigvee_{\beta_i=-Bq}^{Bq} \left(\left[ry_iq \mod q^2\right] = \beta_i\right). \quad (28.7)$$

As this event can only take place whenever β_i is a multiple of q (otherwise, the equality cannot be satisfied), we need only to consider the restricted union of events

$$\bigvee_{\beta_i=-B}^{B} \left(\left[ry_iq \mod q^2\right] = \beta_i q\right) = \bigvee_{\beta_i=-B}^{B} \left([ry_i \mod q] = \beta_i\right). \quad (28.8)$$

Furthermore,

$$C_\ell = \bigvee_{\beta_\ell=-Bq}^{Bq} \left(\left[r \mod q^2\right] = \beta_\ell\right) = \bigvee_{\beta_\ell=-Bq}^{Bq} (r = \beta_\ell) \quad (28.9)$$

which is restricted to $\beta_\ell \in S$ by the last condition. Thus, the probability in Equation (28.6) becomes

$$\mathbb{P}\left(\left(\bigwedge_{i=0}^{\ell-1}\bigvee_{\beta_i=-B}^{B}([ry_i \mod q]=\beta_i)\right)\wedge\left(\bigvee_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq}(r=\beta_\ell)\right)\right). \tag{28.10}$$

Reordering the events gives

$$\mathbb{P}\left(\bigvee_{\beta_0=-B}^{B}\cdots\bigvee_{\beta_{\ell-1}=-B}^{B}\bigvee_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq}\left(\bigwedge_{i=0}^{\ell-1}([ry_i \mod q]=\beta_i)\wedge(r=\beta_\ell)\right)\right). \tag{28.11}$$

As these events are mutually exclusive, this probability is equal to

$$\sum_{\beta_0=-B}^{B}\cdots\sum_{\beta_{\ell-1}=-B}^{B}\sum_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq}\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1}([ry_i \mod q]=\beta_i)\wedge(r=\beta_\ell)\right). \tag{28.12}$$

Using *Bayes' conditional probability rule* [Bay01], this quantity can be rewritten as:

$$\sum_{\beta_0=-B}^{B}\cdots\sum_{\beta_{\ell-1}=-B}^{B}\sum_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq}\mathbb{P}\left(r=\beta_\ell\right)\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1}([ry_i \mod q]=\beta_i)\;\middle|\;(r=\beta_\ell)\right) \tag{28.13}$$

$$=\sum_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq}\mathbb{P}\left(r=\beta_\ell\right)\sum_{\beta_0=-B}^{B}\cdots\sum_{\beta_{\ell-1}=-B}^{B}\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1}([ry_i \mod q]=\beta_i)\;\middle|\;(r=\beta_\ell)\right) \tag{28.14}$$

Naturally $\mathbb{P}\left(r=\beta_\ell\right)=\frac{1}{q^2}$ for any $\beta_\ell\in\mathbb{Z}_{q^2}$. It remains to investigate the value of the second probability in Equation (28.14). To do so, we rewrite $\beta_\ell=g_\ell\beta'_\ell$ where $g_\ell=\gcd(\beta_\ell,q)$. Then, for fixed $\beta_0,\ldots,\beta_{\ell-1},\beta_\ell$, we have

$$\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1}([ry_i \mod q]=\beta_i)\;\middle|\;(r=\beta_\ell)\right) \tag{28.15}$$

$$=\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1}([\beta_\ell y_i \mod q]=\beta_i)\right) \tag{28.16}$$

$$=\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1}(\left[g_\ell\beta'_\ell y_i \mod q\right]=\beta_i)\right) \tag{28.17}$$

The events in this probability are only satisfiable if β_i is a multiple of g_ℓ, say $\beta_i = \beta'_i g_\ell$. Thus, our cumulative probability is rewritten as

$$\sum_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq} \frac{1}{q^2} \sum_{\beta'_0=-\lceil B/g_\ell\rceil}^{\lfloor B/g_\ell\rfloor} \cdots \sum_{\beta'_{\ell-1}=-\lceil B/g_\ell\rceil}^{\lfloor B/g_\ell\rfloor} \mathbb{P}\left(\bigwedge_{i=0}^{\ell-1} ([g_\ell\beta'_\ell y_i \mod q] = \beta'_i g_\ell)\right) \tag{28.18}$$

$$= \sum_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq} \frac{1}{q^2} \sum_{\beta'_0=-\lceil B/g_\ell\rceil}^{\lfloor B/g_\ell\rfloor} \cdots \sum_{\beta'_{\ell-1}=-\lceil B/g_\ell\rceil}^{\lfloor B/g_\ell\rfloor} \mathbb{P}\left(\bigwedge_{i=0}^{\ell-1} \left(\left[\beta'_\ell y_i \mod \frac{q}{g_\ell}\right] = \beta'_i\right)\right) \tag{28.19}$$

By definition $g_\ell = \gcd(\beta_\ell, q)$, which implies that β'_ℓ is invertible modulo $\frac{q}{g_\ell}$. As

$$\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1} \left(\left[\beta'_\ell y_i \mod \frac{q}{g_\ell}\right] = \beta'_i\right)\right) \tag{28.20}$$

$$=\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1} \left(\left[y_i \mod \frac{q}{g_\ell}\right] = \left[\beta'_i\beta'^{-1}_\ell \mod \frac{q}{g_\ell}\right]\right)\right), \tag{28.21}$$

it is clear that each event in the intersection depends only on y_i so that the events are mutually independent. Thus,

$$\mathbb{P}\left(\bigwedge_{i=0}^{\ell-1} \left(\left[y_i \mod \frac{q}{g_\ell}\right] = \left[\beta'_i\beta'^{-1}_\ell \mod \frac{q}{g_\ell}\right]\right)\right) \tag{28.22}$$

$$=\prod_{i=0}^{\ell-1} \mathbb{P}\left(\left[y_i \mod \frac{q}{g_\ell}\right] = \left[\beta'_i\beta'^{-1}_\ell \mod \frac{q}{g_\ell}\right]\right) \tag{28.23}$$

$$=\left(\frac{1}{\frac{q}{g_\ell}}\right)^\ell \tag{28.24}$$

$$=\left(\frac{g_\ell}{q}\right)^\ell. \tag{28.25}$$

Thereby, the cumulative probability is given by

$$\sum_{\substack{\beta_\ell=-Bq\\ \beta_\ell\in S}}^{Bq} \frac{\ell(2\lfloor B/g_\ell\rfloor + 1)}{q^2} \left(\frac{g_\ell}{q}\right)^\ell. \tag{28.26}$$

□

To put the previous lemma into context, we point out that if $S = \{kq \mid k \in \mathbb{Z}\}$, then the probability is smaller than $\frac{(2B+1)\ell}{q^2}$ which is upper bounded by $\frac{1}{\sqrt{q}}$ for sufficiently large q. Thus, although our lattice contains the short vector $\mathbf{v}_1 = (0, \ldots, 0, \frac{q}{g})$ and its multiples, the probability of finding a short vector with a non-zero entry in the first $\ell - 1$ vector entries is rapidly decreasing for increasing q. For small bounds B, we expect with high probability that such a vector does not even exist. So, we may return to our initial guess claiming that the size of the second lattice minima is

$$\lambda_2(\Lambda) \geq \min\left\{q^2, \sqrt{\frac{\ell+1}{8\pi e}} q^{\frac{2\ell}{\ell+1}}\right\}. \tag{28.27}$$

28.3 Case of a (M)iNTRU challenge vector

Assume next that the challenge vector $\mathbf{x} = (x_0, \ldots, x_\ell)$ has been synthetically constructed following the (M)iNTRU distribution. Then, Λ contains, apart from the trivially short vector $v_1 = (0, \ldots, 0, \frac{q}{g})$ and its multiples, another unusually short vector.

28.3.1 An observation

We recall that a synthetically constructed vector $\mathbf{a} := (a_0, \ldots, a_\ell) \in \mathbb{Z}_q^{\ell+1}$ following the (M)iNTRU distribution satisfies

$$a_0 := \left[s^{-1}e_0 \mod q\right] \quad \text{and} \quad a_i := \left[s^{-1}(2^{i-1} - e_i) \mod q\right] \tag{28.28}$$

for all $i \in \{1, \ldots, \ell\}$ where $e_0, \ldots, e_\ell$ denote random errors sampled from the symmetric error distribution χ producing with overwhelming probability elements that are small in absolute value. Thereby, the transformation

$$y_0 := \left[s^{-1}e_0 \mod q\right] \quad \text{and} \quad y_i := \left[s^{-1}(-2e_i + e_{i+1}) \mod q\right] \tag{28.29}$$

for all $i \in \{1, \ldots, \ell - 1\}$ produces elements where the numerators e_0 and $-2e_i + e_{i+1}$ for all $i \in \{1, \ldots, \ell - 1\}$ follow the distribution χ' with mean $\mu_{\chi'} = 0$ and variance $\sigma_{\chi'}^2 = 3\sigma_\chi^2$ so that we can expect them to be quite small when compared to the modulus q.

28.3.2 An unusually short lattice vector

We remark that Λ contains the vector

$$\mathbf{v}_2 := (e_0 q, (-2e_1 + e_2)q, \ldots, (-2e_{\ell-1} + e_\ell)q, s) \tag{28.30}$$

obtained by multiplying the first row in Equation (28.2) by s and subtracting the other rows as often as necessary. Compared to Equation (28.27), this vector is unusually short. Indeed, assume that the error entries are upper bounded by some constant $K > 0$, for example

$$\max\{|e_0|, |-2e_1 + e_2|, \ldots, |-2e_{\ell-1} + e_\ell|\} \leq K. \tag{28.31}$$

Then, the size of v may be upper bounded by

$$\|\mathbf{v}_2\|_2 \leq \sqrt{\ell K^2 q^2 + s^2} \leq \sqrt{\ell K^2 q^2 + q^2} \leq \sqrt{\ell+1}\, qK. \tag{28.32}$$

If K is sufficiently small, then this upper bound is smaller than the expected heuristic size in Equation (28.27). More precisely, we deduce the following proposition.

Proposition 28.2. *If* $\max\{|e_0|, |-2e_1+e_2|, \ldots, |-2e_{\ell-1}+e_\ell|\} \leq \min\left\{\frac{q}{\sqrt{\ell+1}}, \frac{1}{\sqrt{8\pi e}} q^{\frac{\ell-1}{\ell+1}}\right\}$*, then the target vector* $\mathbf{v}_2$ *is shorter than* $\min\left\{q^2, \sqrt{\frac{\ell+1}{8\pi e}} q^{\frac{2\ell}{\ell+1}}\right\}$.

Proof. Replace K in Equation (28.32) by $\min\left\{\frac{q}{\sqrt{\ell+1}}, \frac{1}{\sqrt{8\pi e}} q^{\frac{\ell-1}{\ell+1}}\right\}$ □

As usually the error terms are chosen to be of size $O(\sqrt{q})$, our target vector $\mathbf{v}_2$ is almost surely smaller than the expected heuristic bound in Equation (28.27).

Remark 28.3. *Proposition 28.2 reveals the reason why we multiplied all but one entry of the first row of the lattice basis for Λ in Equation (28.2) by q. The last entry of $\mathbf{v}$ encodes the secret s which might be large compared to the error values. To compensate for its size, we need to increase the size of the error values. As $s < q$, the multiplicand q achieves this goal.*

28.4 Lattice reduction

As the above development yields a motivated heuristic estimate on the expected size of the second shortest lattice vector of Λ, we can investigate the effect of elementary lattice reduction on Λ. To carry out this analysis, we use LLL reduction. We note that an LLL reduced basis $(\mathbf{w}_1, \ldots, \mathbf{w}_{\ell+1})$ satisfies $\|\mathbf{w}_i\|_2 \leq \alpha^{\frac{\ell}{2}} \lambda_i(\Lambda)$ where $\lambda_i(\Lambda)$ denotes the i-th successive minimum of Λ and $\alpha = \frac{1}{\delta - \frac{1}{4}}$. For the sake of explicit results, we consider $\delta = \frac{63}{64} < 0.99$ resulting in $\alpha = \frac{64}{47} < \sqrt{2}$.

Let us start by considering the first LLL reduced vector, namely $\mathbf{w}_1$. As Λ contains the vector $\mathbf{v}_1 = (0, \dots, 0, \frac{q}{g})$ where $g = \gcd(y_0, \dots, y_{\ell-1}, q)$, we assume $\lambda_1(\Lambda) \leq \frac{q}{g}$. This implies that

$$\|\mathbf{w}_1\|_2 \leq \alpha^{\frac{\ell}{2}} \frac{q}{g} \leq 2^{\frac{\ell}{4}} \frac{q}{g} \leq 2^{\frac{\log_2(q)+1}{4}} \frac{q}{g} \leq (2q)^{\frac{1}{4}} \frac{q}{g}. \tag{28.33}$$

Lemma 28.1 yields that such a short vector can only be found with extremely low probability. Thus, it is unlikely that, apart from $\mathbf{v}_1$ and its multiples, another vector of this magnitude exists. Thereby, we can expect $\mathbf{w}_1$ to be (a multiple of) $\mathbf{v}_1$.

28.4.1 Case of a random challenge vector

If $\mathbf{x}$ was chosen uniformly at random, we concluded in Equation (28.27) that the second lattice minima can be expected to be of the size $\lambda_2(\Lambda) \geq \min\left\{q^2, \sqrt{\frac{\ell+1}{8\pi e}} q^{\frac{2\ell}{\ell+1}}\right\}$. Thus, the second LLL reduced basis vector is at least of the same size.

28.4.2 Case of a (M)iNTRU challenge vector

On the other hand, if $\mathbf{x}$ was sampled using the (M)iNTRU distribution, we argue that the second reduced LLL basis vector is smaller than its random counterpart. Assuming that $\mathbf{w}_1$ is a multiple of $\mathbf{v}_0$, the second LLL output $\mathbf{w}_2$ needs to contain at least one nonzero entry on the first ℓ vector entries as otherwise it would not be linearly independent from $\mathbf{w}_1$. By Equation (28.32), we know that $\|\mathbf{v}_2\|_2 \leq \sqrt{\ell+1}\, qK$ where K denotes the maximal error term and Proposition 28.2 shows that our target vector is almost surely smaller than $\min\left\{q^2, \sqrt{\frac{\ell}{8\pi e}} q^{\frac{2\ell}{\ell+1}}\right\}$. If we further reduce the upper bound K, we obtain a similar result for $\mathbf{w}_2$.

Proposition 28.4. *If* $\max\{|e_0|, |-2e_1 + e_2|, ..., |-2e_{\ell-1} + e_\ell|\} \leq \min\left\{\frac{q^{3/4}}{2^{1/4}\sqrt{\ell+1}}, \frac{q^{\frac{3\ell-5}{4(\ell+1)}}}{2^{3/4}\sqrt{\pi e}}\right\}$*, then* $\|\mathbf{w}_2\|_2$ *is smaller than* $\min\left\{q^2, \sqrt{\frac{\ell}{8\pi e}} q^{\frac{2\ell}{\ell+1}}\right\}$.

Proof. Let $\max\{|e_0|, |-2e_1 + e_2|, \dots, |-2e_{\ell-1} + e_\ell|\} \leq K$. Then we know that $\|\mathbf{w}_2\|_2 \leq \alpha^{\frac{\ell}{2}} \lambda_2(\Lambda) \leq \alpha^{\frac{\ell}{2}} \mathbf{v}$. Using again the fact that $\alpha \leq \sqrt{2}$ implies $\alpha^{\frac{\ell}{2}} \leq (2q)^{\frac{1}{4}}$ and using Equation (28.32), we conclude

$$\|\mathbf{w}_2\|_2 \leq (2q)^{\frac{1}{4}} \sqrt{\ell+1}\, qK.$$

Replacing K by $\min\left\{\frac{q^{3/4}}{2^{1/4}\sqrt{\ell+1}}, \frac{q^{\frac{3\ell-5}{4(\ell+1)}}}{2^{3/4}\sqrt{\pi e}}\right\}$ proves the proposition. □

The upper bound for K in Proposition 28.4 is $O\left(q^{\frac{3}{4}}\right)$ and, as usually $K = O(\sqrt{q})$, we can expect the condition to hold in practice. In comparison, using Lemma 28.1, we conclude again that such a short vector can only be expected with low probability.

28.5 Conclusion

We conclude that:

1. If the challenge vector was sampled uniformly at random, then the second LLL reduced vector can be expected to be lower bounded by $\min\left\{q^2, \sqrt{\frac{\ell}{8\pi e}}q^{\frac{2\ell}{\ell+1}}\right\}$.
2. If the challenge vector follows the (M)iNTRU distribution, then the second LLL reduced vector is with high probability smaller than $\min\left\{q^2, \sqrt{\frac{\ell}{8\pi e}}q^{\frac{2\ell}{\ell+1}}\right\}$.

Hence, we can heuristically distinguish between a uniformly at random sampled challenge vector and a synthetically constructed one by simply comparing the length of the second LLL reduced vector with $\min\left\{q, \sqrt{\frac{\ell}{8\pi e}}q^{\frac{2\ell}{\ell+1}}\right\}$.

28.6 Empirical tests

An implementation of this distinguisher in SAGEMATH (distinguisher2) and the corresponding statistics can be found at:

https://orbilu.uni.lu/handle/10993/47990

The practical experiments confirm the heuristic assumption on the first and second LLL reduced basis vector, as well as the subsequent conclusions. Furthermore, in case of a (M)iNTRU challenge vector, it seems that we can filter out the secret s. Indeed, typically, the second LLL reduced vector $\mathbf{w}_2$ corresponds to $\pm\mathbf{v}_2$ and its last entry directly reveals s. Thus, the corresponding (M)iNTRU search problem seems to be solvable with essentially the same lattice method. However, a theoretical confirmation of the underlying heuristic assumptions is still open.

Chapter 29

Generalizations and limitations of our attacks

After mounting two attacks against the (M)iNTRU decision assumption, we quickly discuss how the attacks can be generalized to its parent assumptions. We exemplify this generalization on the MiNTRU assumption in Section 29.1, the generalization to the general iNTRU assumption being similar. We note that these generalizations are strictly limited and do not endanger cryptographic instantiations yet. These limitations are discussed in detail in Section 29.2.

29.1 MiNTRU

Intuitively, replacing the one-dimensional integer elements in the (M)iNTRU problem with matrices leads to the MiNTRU problem. Thus, it is not surprising that our attacks can be modified to apply to the MiNTRU decision problem. Hereinafter, we develop an attack based on our first distinguisher. The development is based on full-length challenge matrices, but can be adapted to shortened matrices as well.

29.1.1 Lattice construction

Let $\mathbf{X}$ be a challenge matrix following either the uniform distribution or the MiNTRU distribution over $\mathbb{Z}_q^{n\times m}$ where $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ and $m = n(\ell+1)$. We decompose our challenge matrix into $(\ell+1)$ individual $n \times n$ matrices $\mathbf{X}_0, \ldots, \mathbf{X}_\ell$ such that

$$\mathbf{X} = [\mathbf{X}_0|\ldots|\mathbf{X}_\ell]. \tag{29.1}$$

Then, we construct the matrix $\mathbf{Y} = [Y_0| \dots |Y_{\ell-1}]$ by setting

$$\mathbf{Y}_0 := [\mathbf{X}_0 \mod q] \quad \text{and} \quad \mathbf{Y}_i = [2\mathbf{X}_i - \mathbf{X}_{i+1} \mod q] \tag{29.2}$$

for all $i \in \{1, \dots, \ell-1\}$ where the modulo operation returns for each matrix entry the unique representative of the result in $\mathbb{Z}_q$. This allows us to define the $\ell n \times \ell n$ q-ary row lattice

$$\Lambda_q(\mathbf{Y}) = \left\{ \mathbf{t} \in \mathbb{Z}^{\ell n} \;\middle|\; \mathbf{t} \equiv \mathbf{y}\mathbf{Y} \mod q \text{ for some } \mathbf{y} \in \mathbb{Z}^n \right\}. \tag{29.3}$$

Remark 29.1. *As in the one-dimensional case, we remark that almost surely there is one $n \times n$ submatrix of $\mathbf{Y}$ that is invertible modulo q. Indeed, let the prime decomposition of q be given as $q = \prod_{i=1}^{k} q_i^{\alpha_i}$. Then, a random $n \times n$ matrix is invertible modulo q if and only if it is so modulo $q_i^{\alpha_i}$ for all $i \in \{1, \dots, k\}$, which is valid for a single i with probability $\prod_{j=1}^{n}(1 - q_i^{-j\alpha_i})$. Therefore, the probability that none of the matrices $\mathbf{Y}_0, \dots, \mathbf{Y}_{\ell-1}$ is invertible is only $\left(1 - \prod_{i=1}^{k} \prod_{j=1}^{n}(1 - q_i^{-j\alpha_i})\right)^{\ell}$.*

Assume that there is an index $t \in \{0, \dots, \ell-1\}$ such that the matrix $\mathbf{Y}_t$ is invertible modulo q. Then, we can define

$$\mathbf{Z}_i := \left[\mathbf{Y}_t^{-1}\mathbf{Y}_i \mod q\right] \quad \forall i \in \{0, \dots, \ell-1\} \tag{29.4}$$

where $\mathbf{Z}_t = \mathbf{I}$ is the $n \times n$ identity matrix. We note that

$$\Lambda_q(\mathbf{A}) = \mathcal{L}\begin{pmatrix} \mathbf{Z}_0 & \dots & \mathbf{Z}_{t-1} & \mathbf{I} & \mathbf{Z}_{t+1} & \dots & \mathbf{Z}_{\ell-1} \\ q\mathbf{I} & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \dots & q\mathbf{I} & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & q\mathbf{I} & \dots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & q\mathbf{I} \end{pmatrix} \tag{29.5}$$

and find a direct generalization of Lemma 27.3.

29.1.2 Case of a random challenge matrix

Assume that our initial challenge matrix $\mathbf{X}$ was sampled uniformly at random. Then, our constructed matrices $\mathbf{Y}_i$ still follow the uniform distribution. By Theorem 7.5,

$$\mathbb{P}\left(\lambda_1(\Lambda_q(\mathbf{Y})) \geq \min\left\{q, \sqrt{\frac{n\ell}{8\pi e}} q^{\frac{\ell-1}{\ell}}\right\}\right) \geq 1 - \frac{1}{\sqrt{\pi n \ell}} 2^{-n\ell} \tag{29.6}$$

where $\lambda_1(\Lambda_q(\mathbf{Y}))$ denotes the first lattice minimum of $\Lambda_q(\mathbf{Y})$ in the Euclidean norm. Thereby, the shortest lattice vector can be expected to satisfy

$$\lambda_1(\Lambda_q(\mathbf{Y})) \geq \min\left\{q,\ \sqrt{\frac{n\ell}{8\pi e}} q^{\frac{\ell-1}{\ell}}\right\}. \tag{29.7}$$

29.1.3 Case of a MiNTRU challenge matrix

Assume next that the initial challenge matrix $\mathbf{X}$ was synthetically constructed following the MiNTRU distribution. We will show that in this case, the lattice contains a nontrivial short vector, magnitudes smaller than the heuristic in Equation (29.7).

First observation

We recall that a synthetically constructed matrix $\mathbf{A}$ following the MiNTRU distribution satisfies $\mathbf{A} := \left[\mathbf{S}^{-1} \times (\mathbf{G} - \mathbf{E}) \mod q\right]$. Thus, decomposing $\mathbf{A} = [\mathbf{A}_0|\dots|\mathbf{A}_\ell]$ yields

$$\begin{aligned} \mathbf{A}_0 &= \left[\mathbf{S}^{-1}(\mathbf{G}_0 - \mathbf{E}_0) \mod q\right] \quad \text{and} \\ \mathbf{A}_i &= \left[\mathbf{S}^{-1}(2^{i-1}\mathbf{I} - \mathbf{E}_i) \mod q\right] \end{aligned} \tag{29.8}$$

for all $i \in \{1, \dots, \ell\}$ where $\mathbf{E}_0, \dots, \mathbf{E}_\ell$ denote random error matrices whose entries are sampled from the symmetric error distribution χ producing with overwhelming probability elements that are small in absolute value. Thereby,

$$\begin{aligned} \mathbf{Y}_0 &= \left[-\mathbf{S}^{-1}\mathbf{E}_0 \mod q\right] \equiv \mathbf{S}^{-1}\mathbf{E}'_0 \mod q \quad \text{and} \\ \mathbf{Y}_i &= \left[\mathbf{S}^{-1}(-2\mathbf{E}_i + \mathbf{E}_{i+1}) \mod q\right] \equiv \mathbf{S}^{-1}\mathbf{E}'_i \mod q \end{aligned} \tag{29.9}$$

for all $i \in \{1, \dots, \ell - 1\}$ where the entries of $\mathbf{E}'_i$ follow the distribution χ' with mean $\mu_{\chi'} = 0$ and standard deviation $\sigma' = \sqrt{3}\sigma$. In particular, as χ produces with overwhelming probability elements with a small absolute value, so does χ'.

Second observation

Continuing to outline the effect of our variable changes leads to

$$\mathbf{Z}_i = \left[\mathbf{E}'^{-1}_t \mathbf{E}'_i \mod q\right] \quad \forall i \in \{0, \dots, \ell - 1\}. \tag{29.10}$$

Third observation

By construction, it is clear that

$$\mathbf{E}_t'\mathbf{Z}_i \equiv \mathbf{E}_i' \mod q \quad \forall i \in \{0, \ldots, \ell - 1\} \tag{29.11}$$

and so, denoting the j-th row of $\mathbf{E}_i'$ by $\mathbf{e}_i^{(j)}$ yields

$$\mathbf{e}_t^{(j)}\mathbf{Z}_i \equiv \mathbf{e}_i^{(j)} \mod q \quad \forall i \in \{0, \ldots, \ell - 1\} \text{ and } \forall j \in \{1, \ldots, n\}. \tag{29.12}$$

Some short lattice vectors

Interestingly, our observations imply that our lattice contains the n linearly independent vectors

$$\mathbf{e}^{(j)} = \left(\mathbf{e}_0^{(j)}| \ldots |\mathbf{e}_\ell^{(j)}\right) \quad \forall j \in \{1, \ldots, n\} \tag{29.13}$$

obtained through a linear combination (defined by $\mathbf{e}_t^{(j)}$) of the first n rows followed by subtracting the other rows as often as needed. Assume that the error entries of the error matrices $\mathbf{E}_i'$ (or only of a specific row of the error matrices) are upper bounded by some constant $K > 0$. Then, the size of $\mathbf{e}^{(j)}$ is upper bounded by

$$\|\mathbf{e}^{(j)}\|_2 \leq \sqrt{\ell n K^2} \leq \sqrt{\ell n}\, K. \tag{29.14}$$

Proposition 29.2. *If the error entries of the error matrices* $\mathbf{E}_i'$ *are upper bounded by* $\min\left\{\frac{q}{\sqrt{\ell n}}, \frac{1}{\sqrt{8\pi e}}q^{(\ell-1)/\ell}\right\}$*, then the target vectors* $\mathbf{e}^{(j)}$ *are shorter than* $\min\left\{q,\ \sqrt{\frac{\ell n}{8\pi e}q^{\frac{\ell-1}{\ell}}}\right\}$ *for each* $j \in \{1, \ldots, n\}$.

Proof. Replace K in Equation (29.14) by $\min\left\{\frac{q}{\sqrt{\ell n}}, \frac{1}{\sqrt{8\pi e}}q^{(\ell-1)/\ell}\right\}$. □

Since in practice the error terms are of size $O(\sqrt{q})$, the size condition is almost always satisfied.

29.1.4 Lattice reduction

Having good estimates on the expected size of the shortest lattice vector, we can investigate the effect of lattice reduction.

Case of a random challenge matrix

If $\mathbf{X}$ was sampled uniformly at random, we concluded in Equation (29.7) that the shortest vector can be expected to be of size $\lambda_1(\Lambda_q(\mathbf{Y})) \geq \min\left\{q, \sqrt{\frac{n\ell}{8\pi e}} q^{\frac{\ell-1}{\ell}}\right\}$ and so any reduced vector is at least of the same size.

Case of a MiNTRU challenge matrix

Contrary to the one-dimensional case, the large lattice dimension avoids the theoretic conclusion of an unusually short lattice vector through usual LLL reduction. Indeed, LLL with $\delta = \frac{63}{64}$ outputs $\mathbf{w}_1$ such that

$$\|\mathbf{w}_1\|_2 \leq q^{\frac{n}{4}} 2^{\frac{n-1}{4}} \sqrt{\ell n} K \tag{29.15}$$

which is too loose for a theoretical conclusion. In particular, if $n \geq 4$, the bound is larger than q, the trivial upper bound for the lattice minima in a q-ary lattice.

Remark 29.3. *Note that if $n = 1$, then we recover the same bound as in Equation* (27.14)*.*

Since in general LLL performs better in practice than in theory, good results can be expected for low degrees (we achieved a 50% success rate for dimension 6), but a general solution cannot be expected in this direction.

Through BKZ reduction, the approximation factor can be strongly improved and for a small enough block size, one may even get a polynomial runtime (see [LN20, Theorem 2]). Thus, slightly larger dimensions may be treated. To be precise, BKZ with block size β achieves

$$\|\mathbf{b}_1\|_2 \leq \gamma_\beta^{\frac{\ell n-1}{\beta-1}} \lambda_1(\Lambda_q(\mathbf{Y})) \tag{29.16}$$

for its first reduced vector $\mathbf{b}_1$ where γ_β denotes the Hermite constant. To put this into context, consider the block size $\beta = 24$ for which BKZ still finishes in a reasonable amount of time. Then,

$$\|\mathbf{b}_1\|_2 \leq 4^{\frac{\ell n-1}{23}} \lambda_1(\Lambda_q(\mathbf{Y})) \leq 2^{\frac{2\ell n}{23}} \sqrt{\ell n} K \leq (2q)^{\frac{2n}{23}} \sqrt{\ell n} K \tag{29.17}$$

indicating that the method would work for sufficiently small dimension n and noise K. Unfortunately, this improvement is still not exact enough to handle large dimensions. The same holds if better BKZ bounds are used for the cryptanalysis (see [GN08, SB10]).

A speculative dimension gain might be achieved through the new BKZ 2.0 algorithm [CN11] which is particularly well suited for large block sizes. Nonetheless, the inherent size condition and the time increase remain. Theoretical success can only be shown through exact shortest vector problem solvers whose run-times are exponential in the lattice dimension. Therefore, this attack is impractical for large dimensions.

29.1.5 Conclusion

We conclude that:

1. If the challenge matrix was sampled uniformly at random, then the shortest lattice vector can be expected to be lower bounded by $\min\left\{q,\ \sqrt{\frac{\ell n}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$.

2. If the challenge matrix follows the MiNTRU distribution, then the shortest lattice vector is, conditioned on the error distribution χ, smaller than $\min\left\{q,\ \sqrt{\frac{\ell n}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$.

Hence, if we could achieve a tight approximation of the shortest lattice vector, we could distinguish with high probability between a randomly sampled challenge matrix and a synthetically constructed MiNTRU matrix by simply comparing the length of the shortest vector approximation with $\min\left\{q,\ \sqrt{\frac{\ell n}{8\pi e}}q^{\frac{\ell-1}{\ell}}\right\}$. An implementation of the corresponding distinguisher can be found at:

https://orbilu.uni.lu/handle/10993/47990

29.1.6 Improvements and observations

Remark 1: Recovering the secret

We note that if a MiNTRU matrix is detected, then it is tempting to pad the n shortest lattice vectors together to recover the error matrices $\mathbf{E}_i'$, which can be used to reveal first the original error matrices $\mathbf{E}_i$ and subsequently the secret matrix $\mathbf{S}$. Unfortunately, such a procedure does not work in general. Indeed, the shortest lattice vectors may not correspond to the error vectors, as shorter vectors may be obtained through linear combinations of them.

Remark 2: Minor improvement

To reduce computational power, it is crucial to decrease the lattice dimension. This may be achieved by not considering the ℓ matrices $\mathbf{Z}_0, \ldots, \mathbf{Z}_{\ell-1}$, but only b of them for some $1 \leq b \leq \ell$. As long as the error terms are small enough (c.f. Proposition 29.2), the construction works out. This can also be formally proven by simply replacing ℓ with b in the whole chapter. We note that it doesn't matter which copies $\mathbf{Z}_i$ are chosen as long as $\mathbf{Z}_t$ is part of the chosen set. Intuitively, $\mathbf{Z}_t$ guarantees that only small linear combinations appear after reduction. The other $b-1$ $\mathbf{Z}_i$'s may be chosen at random. Empirical tests suggested that $b = 5$ is sufficient for the standard deviation $\sigma_\chi = 2\sqrt{q}$. Furthermore, we note that there is no restriction on choosing complete copies of the matrices $\mathbf{Z}_i$. Any combination of individual columns works out fine. Only $\mathbf{Z}_t$ must appear completely. However, this improvement does not solve the real bottleneck of the attack, which is the large matrix dimension n.

Remark 3: Dependence of matrices

When developing the matrices $\mathbf{Z}_i = \left[\mathbf{Y}_t^{-1}\mathbf{Y}_i \mod q\right]$, one may think of repeating the same construction for another index t' in order to obtain linearly independent matrices $\mathbf{Z}_i' = \left[\mathbf{Y}_{t'}^{-1}\mathbf{Y}_i \mod q\right]$ that can be used to gain more knowledge on the corresponding lattices. Unfortunately, this is not the case as $\mathbf{Z}_i' \equiv \mathbf{Z}_t'\mathbf{Z}_i \mod q$ and so no more information can be generated through a simple index change.

Remark 4: iNTRU

A similar development shows that the attack may be adapted to the general iNTRU assumption. However, it suffers from the same restrictions. The dimension of the considered polynomial ring or equivalently the degree of the reduction polynomial takes on the role of the matrix dimension showing that the attack only works for small degrees.

29.2 Impact on cryptographic schemes

Our attacks are devised for the case of integer rings only. Their efficiency for general rings is limited. Although they represent a potential theoretical threat, our constructions are not strong enough to impact the security of concrete cryptographic constructions such as [GL20] or [GGH+19] that use iNTRU with rings of a large degree or MiNTRU with matrices of a large dimension. For example, [GGH+19] suggests to use the MiNTRU assumption with matrix dimension $n = O\left(q^{\frac{1}{4}}\right)$ where q, in turn, is of size $q = O(2^{42})$. Further security analyses may be required to develop practical attacks against protocols using the iNTRU and MiNTRU assumption with cryptographic parameters. Our development gives a first indication that the iNTRU, respectively the MiNTRU assumption, may not be as hard as other well-known assumptions such as LWE, but it does not have a direct impact on cryptographic instantiations.

Other such doubts were raised by *Lee* and *Wallet* in [LW20] who developed a sublattice attack based on [KF17] against the MiNTRU assumption. Albeit the authors manage to fully retrieve secret Bernoulli matrices $\mathbf{S} \in \{-1, 0, 1\}^{n \times m}$ from a given MiNTRU matrix, the manageable entropic noise is limited. We note that [GGH+19] was updated to bypass the toy example attack from [LW20] by setting the secret matrix to not being a Bernoulli matrix anymore. However, the claimed hardness of the underlying assumption is debatable. In Chapter 30, we deepen this suspicion by comparing the MiNTRU to other well-known hardness assumptions.

Chapter 30

Comparisons with other hardness assumptions

To position the iNTRU and MiNTRU problems in the cryptographic field, we compare them to some related computational hardness assumptions and outline some inherent differences.

30.1 NTRU

Given the nominative link to the famous NTRU cryptosystem [HPS98], one expects that the iNTRU and MiNTRU problems can be obtained as variants of the former well-established counterpart. Naturally, we observe the similarities of iNTRU's use of polynomial rings with the NTRU instantiation, but iNTRU misses the intricate polynomial convolution and the double modulus of NTRU. In [GGH$^+$19, Section 1.2], the authors presented a connection of iNTRU to Matrix-NTRU and in [GGH$^+$19, Appendix D], the small secret MiNTRU variant is cryptanalyzed by known NTRU attacks. The *small secret* variant of MiNTRU is obtained by first sampling the secret matrix $\mathbf{S} \in (\mathbb{Z}_q)^{n\times n}_{inv}$ such that its entries follow the error distribution χ, and then removing the first $n \times n$ block from the MiNTRU matrix. The authors considered known NTRU attacks such as dimension reduction [MS01], the hybrid attack [HG07], and the overstretched key recovery attack [KF17] to assess the hardness of the small secret MiNTRU variant. We note that [LW20] used the overstretched attack against Bernoulli secrets (see Section 29.2). However, apart from the attacks, the connection to NTRU and its innate security level is missing.

30.2 Learning-with-Errors

Due to the link of the iNTRU and MiNTRU problem with lattices, it is natural to consider other lattice-related primitives for comparison purposes. Besides the NTRU problem, we may consider the *learning-with-errors* (LWE) [Reg05, Reg10] problem. Roughly speaking, the LWE search problem over finite rings first chooses a secret vector $\mathbf{s} \in \mathbb{Z}_q^n$, samples a vector $\mathbf{a} \in \mathbb{Z}_q^n$ uniformly at random and samples an error element $e \in \mathbb{Z}_q$ from a particular error distribution χ. Subsequently, it computes $t = [\langle \mathbf{a}, \mathbf{s} \rangle + e \mod q]$ and asks to retrieve $\mathbf{s}$ from $(\mathbf{a}, t)$. Usually, not a single sample, but a polynomial number of samples $(\mathbf{a}, t)$ is given, with varying $\mathbf{a}$ and e. For our comparison, we consider a particular variant of the general LWE problem.

30.2.1 Multi-secret LWE

The *multi-secret* variant of the learning-with-errors problem is particularly well suited to be compared with the MiNTRU problem. The multi-secret variant was used in [PW08, PVW08, GGH+19] and can be reduced to the usual LWE. It is defined by a particular sampling process:

Let $q \in \mathbb{Z}_{\geq 2}$, $n \in \mathbb{Z}_{\geq 1}$, $m \in \mathbb{Z}_{\geq n}$, and χ be a (symmetric) error distribution over $\mathbb{Z}_q$ with standard deviation $\sigma_\chi = O(\sqrt{q})$.

Definition 30.1 (Multi-secret LWE distribution)**.** Let $\mathbf{S} \in \mathbb{Z}_q^{n \times n}$ be sampled uniformly at random and let $\mathbf{E} \in \mathbb{Z}_q^{n \times m}$ be sampled such that its entries follow the error distribution χ. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be sampled uniformly at random. Set the multi-secret LWE distribution as the distribution of pairs $(\mathbf{A}, \mathbf{B})$ where

$$\mathbf{B} := [\mathbf{S} \times \mathbf{A} + \mathbf{E} \mod q] \, . \tag{30.1}$$

The multi-secret LWE search problem asks to retrieve the secret matrix $\mathbf{S}$ from a multi-secret LWE pair $(\mathbf{A}, \mathbf{B})$ and its modulus q. The corresponding decision problem asks to distinguish a multi-secret LWE pair $(\mathbf{A}, \mathbf{B})$ from a pair $(\mathbf{X}, \mathbf{Y})$ that was sampled uniformly at random. It is believed that both problems can only be solved with negligible probability.

30.2.2 Hardness of multi-secret LWE

The hardness of LWE problems has been well-studied and many surprising properties, such as self-reducibility [Reg05] proving hardness on average, were found. Furthermore, explicit reductions to known supposedly hard lattice problems were outlined [Reg05, Pei09]. Those results and the fact that after 16 years, no efficient general attack was mounted push us to believe that the LWE problems are hard.

However, not all LWE instances are equally hard. Indeed, their hardness depends on the chosen error scale χ. For example, neither the decision nor the search variant is solvable for completely random errors, but are solvable for (binary) Bernoulli errors [STA20]. The number of samples obtained for an LWE instance seems to have only a small impact on the hardness as one can produce, through linear combinations, as many valid samples as required [Reg10]. Yet, most hardness reductions depend on sample restrictions [STA20].

Multi-secret LWE can be reduced to the usual LWE problem and the additional information leakage (of the multiple secrets) is supposed to not make the problem easier [PW08, Lemma 6.2]. Thus, the multi-secret variant seems to be as hard as the usual LWE.

30.2.3 Connection to MiNTRU

To strengthen the MiNTRU assumption and so the semantic security of their encryption scheme, the original formalization paper [GGH+19] relates the MiNTRU problem to the multi-secret variant of the LWE problem. Indeed, setting in Definition 30.1 $\mathbf{B} = \mathbf{G}$ and isolating $\mathbf{A}$ leads to the definition of $\mathbf{A}$ in Equation (26.3). Pseudorandomness of MiNTRU in dimension n can be deduced from pseudorandomness of the n-secret LWE. However, this reduction is conditioned on two controversial assumptions. First, the MiNTRU error distribution is distorted because it replaces the natural error distribution $\chi^{n\times m}$ by $\chi^{n\times m} \cdot \mathbf{B}^{-1}(\mathbf{G})$. The latter distribution may not correspond to the low-norm error distribution expected in the MiNTRU setting. In particular, heavier entropic noise is produced. Second, the n-secret LWE is given trapdoor oracle access to $\mathbf{B}$. This trapdoor oracle access consists in a non-standard assumption and its pseudorandomness requires further investigation. The use of the gadget matrix $\mathbf{G}$ in the MiNTRU case and the inversion of $\mathbf{S}$ do apparently not have a considerable effect on the hardness, but, intuitively, they leak some more information on the MiNTRU matrices.

The main comparison point of the multi-secret LWE and MiNTRU stems from our own development. We formally proved that an elementary lattice attack is sufficient to break low-dimensional MiNTRU decision problems. On the contrary, the multi-secret LWE seems to not suffer from such a vulnerability. The most decisive difference can be found in the basic case $n = 1$ where the MiNTRU restricts to (M)iNTRU and the multi-secret LWE restricts to ordinary LWE. On the one hand, our study indicates that, with high probability, the (M)iNTRU decision problem with errors of size $O(\sqrt{q})$ can be solved. Furthermore, it seems that the search version can often be obtained at no additional cost. On the other hand, [APS15], its corresponding LWE estimator, and [BLP+13] show that neither the ordinary search LWE, nor its decision version are solvable if errors of size $O(\sqrt{q})$ are used. This difference may or may not have an impact on the hardness assumption for iNTRU with high degree polynomial rings or MiNTRU with high dimensional matrices, but it certainly outlines a discrepancy between the required secure parameter bounds. After all, multiplying a secret and adding an error is not the same as adding an error and multiplying a secret!

Chapter 31

Open Questions

The iNTRU and MiNTRU assumptions are relatively new and so their hardness is not well understood yet. Although our attack targets to break the MiNTRU decision assumption in low dimensions, we do not advocate abolishing the assumptions in general. Indeed, it is scientifically more valuable to further investigate the hardness assumptions.

On the one hand, new attacks for large dimensions may be devised to further weaken the assumptions. Merging our ideas with those from [LW20] may result in such an attack, but we suppose that the large lattice dimension decreases the general application range. From another perspective, working out the exact distribution of MiNTRU matrices may reveal a computationally detectable difference between the MiNTRU distribution and the uniform distribution.

On the other hand, we note that proving the MiNTRU assumption to be hard in general offers a new pragmatic hardness assumption that can be used for various security proofs. Such a proof may be obtained by reducing the learning-with-errors problem or another supposedly hard problem to the MiNTRU problem without relying on auxiliary non-standard assumptions. Alternatively, the MiNTRU assumption can be strengthened through additional heuristic arguments in favour of its hardness, or through more experimental results on known attacks such as those outlined in [GGH$^+$19].

We take the opportunity to highlight again the key result on q-ary lattices that enabled our analysis in Chapter 27 and Chapter 29, namely Theorem 7.5. This explicit probability result allows us to predict the precise success probability and to give exact conditions on when our attack applies. One may formulate similar results for other ℓ^p norms that can be applied to other developments.

Bibliography IV

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 99–108. Association for Computing Machinery, 1996.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. Cryptology ePrint Archive, Report 2015/046, 2015. https://ia.cr/2015/046.

[Bay01] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Rev. R. Acad. Cienc. Exactas Fís. Nat. (Esp.)*, 95(1-2):61–80, 2001. Translated from the English original [R. Soc. Lond. Philos. Trans. **53** (1763), 370–418] by Miguel Ángel Gómez Villegas, Francisco Javier Girón González-Torre, María Lina Martínez García and David Ríos Insua.

[BLP$^+$13] Zvika Brakerski, Adeline Langlois, Chris Peikert, Oded Regev, and Damien Stehlé. Classical hardness of learning with errors. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 575–584, New York, NY, USA, 2013. Association for Computing Machinery.

[CN11] Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology – ASIACRYPT 2011*, pages 1–20, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[Gau66] Carl Friedrich Gauss. *Disquisitiones Arithmeticae.* Yale University Press, New Haven, Conn.-London, 1966. Translated into English by Arthur A. Clarke, S. J.

[GGH$^+$19] Nicholas Genise, Craig Gentry, Shai Halevi, Baiyu Li, and Daniele Micciancio. Homomorphic encryption for finite automata. In *Advances in Cryptology – ASIACRYPT 2019*, LNCS, pages 473–502. Springer, December 2019.

[GL20] Nicholas Genise and Baiyu Li. Gadget-based iNTRU lattice trapdoors. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology–INDOCRYPT 2020*, pages 601–623. Springer, 2020.

[GN08] Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 31–51. Springer, 2008.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, STOC '08, page 197–206. ACM, 2008.

[HG07] Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 150–169, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe P. Buhler, editor, *Algorithmic Number Theory*, Third International Symposium, June 21-25, 1998, pages 267–288. Springer, 1998.

[KF17] Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched NTRU parameters. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 3–26. Springer, 2017.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Math. Comput.*, 48:203–209, 1987.

[Lag13] Jeffrey C. Lagarias. Euler's constant: Euler's work and modern developments. *Bull. Amer. Math. Soc. (N.S.)*, 50(4):527–628, 2013.

[LN20] Jianwei Li and Phong Q. Nguyen. A complete analysis of the BKZ lattice reduction algorithm. Cryptology ePrint Archive, Report 2020/1237, 2020. https://ia.cr/2020/1237.

[LS14] Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75, 06 2014.

[LW20] Changmin Lee and Alexandre Wallet. Lattice analysis on MiNTRU problem. Cryptology ePrint Archive, Report 2020/230, 2020. https://ia.cr/2020/230.

[Mil85] Victor S. Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[MS01] Alexander May and Joseph H. Silverman. Dimension reduction methods for convolution modular lattices. In Joseph H. Silverman, editor, *Cryptography and Lattices*, pages 110–125, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[NIS17] U.S. National Institute of Standards and Technology NIST. Post-Quantum Cryptography. https://csrc.nist.gov/Projects/post-quantum-cryptography, 2017. Last accessed on 04.04.2022.

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 333–342. Association for Computing Machinery, 2009.

[PR06] Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography Conference*, volume 3876 of *LNCS*, pages 145–166. Springer, 2006.

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 554–571, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 187–196. Association for Computing Machinery, 2008.

[RAD78] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. On data banks and privacy homomorphisms. *Foundation of Secure Computations*, 1978.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 84–93. Association for Computing Machinery, 2005.

[Reg10] Oded Regev. The learning with errors problem (invited survey). In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 191–204, 2010.

[RS59] Michael O. Rabin and Dana Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[RS62] J. Barkley Rosser and Lowell Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1):64 – 94, 1962.

[SB10] Michael Schneider and Johannes A. Buchmann. Extended lattice reduction experiments using the BKZ algorithm. In *Sicherheit*, 2010.

[Sho94] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[STA20] Chao Sun, Mehdi Tibouchi, and Masayuki Abe. Revisiting the hardness of binary error LWE. In *Information Security and Privacy: 25th Australasian Conference, ACISP 2020, Perth, WA, Australia, November 30 – December 2, 2020, Proceedings*, page 425–444, Berlin, Heidelberg, 2020. Springer-Verlag.

Part V

A Conditional Attack Against Functional Encryption Schemes

Act V: The cryptographer

Pitying his pops, Jay decided to join him after dinner. Slowly he climbed up the spiral staircase to the attic where his pops has his workspace. After knocking, he opened the croaking wooden door and entered the dusty room. “Pops, are you here”, he asked anxiously. The wind blew softly between the roof boards and a neon tube brightened the room. “Yes, I am reading”, replied his pops calmly, “did you know that the 1933 Double Eagle has been sold for 7.6 million U.S. dollars?”. His pops always followed such sales. “Really?” asked Jay excitedly and approached the rocking chair. “But that’s probably not why you reach out to me, is it?”, questioned pops.

“No”, replied Jay timidly. “Sit down and tell me what lies in your heart?”, invited pops. Jay sat down and asked: “Is it true that you know how to encrypt messages?” “Indeed”, replied pops, “my good friend Alan taught me how to do that. At first, it was a hurdle, and I made many mistakes, but now I believe to have mastered the essentials.” Pops saw the astonished gaze of his grandson and suggested: “Do you want to see my new idea?” Of course, Jay wanted to see it, he might be the first to do so.

Pops took a large notebook out of the closet. It contained a collection of formulas and schematic drawings. “Look here”, pointed out pops, “I invented a way to encrypt our counting list such that our collection can be evaluated but the individual counts are hidden.” He wrote down a particular formula and continued: “I can enable you to find the total of the face values of our coins or the numismatic values, but you will not be able to guess the number of coins without any additional information.” He highlighted a particular part of his formula and indicated: “Although I succeeded in devising such an encryption method, it seems to have some deficiencies. Its security leaves the traditional cryptanalysis and I don’t know whether I captured all security threats.”

Abstract V

Functional encryption emerged as an ambitious cryptographic paradigm generalizing a variety of public-key encryption primitives. Contrary to the classical framework that only permits encrypting or decrypting messages, functional encryption allows evaluating functions over encrypted messages without leaking information on the corresponding plaintext messages. Such powerful machinery has diverse applications ranging from data mining to obfuscation but suffers from severe limitations regarding flexibility and security. Due to its novelty, many of its aspects are not well understood yet and require more research. We develop a conditional attack against a family of functional encryption schemes. We base our attack on the currently weakest data-privacy notion for functional encryption: indistinguishability. Intuitively, indistinguishability in the public-key setting is based on the premise that no adversary can distinguish between the encryptions of two known plaintext messages. As functional encryption allows us to evaluate functions over encrypted messages, the adversary is restricted to evaluations resulting in the same output only. To ensure consistency with other primitives, the decryption procedure of a functional encryption scheme is allowed to fail and output an error. We observe that an adversary may exploit the special role of these errors to craft challenge messages that can be used to win the indistinguishability game. Indeed, albeit the functional evaluation of those messages leads to the common error symbol, their intermediate computation values may differ. A formal decomposition of the underlying functionality into a mathematical function and an error trigger reveals this dichotomy. Our observation yields a practical criterion for the security of functional encryption schemes which leaves the scope of traditional public-key cryptanalysis. To further motivate our abstract attack, we outline its impact on multiple candidate DDH-based inner-product functional encryption schemes when restricting them to bounded-norm evaluations. Finally, we show that a weaker indistinguishability notion that declares those schemes as secure for bounded-norm evaluations is incompatible with other classical primitives.

Contents V

Act V **243**

Abstract V **244**

Contents V **245**

32 A short review of functional encryption **247**

32.1 Historical motivation 247

32.2 Applications 248

32.3 Instantiations 248

33 Functional encryption **249**

34 Security notions **251**

34.1 Data-privacy 251

34.2 Function-hiding 252

34.3 Consistency and robustness 252

35 Focus on data-privacy **253**

35.1 Indistinguishability 253

35.2 Semantic security 255

35.3 Informal comparison 255

36 Our attack **257**

36.1 Functionalities and their error symbols 258

36.2 Functional encryption and error triggers 258

36.3 Revision of indistinguishability 259

36.4 Attacking indistinguishability 260

36.5 Summary 262

37 Impact on the bounded-norm inner-product functionality 263

37.1 Bounded-norm inner-product functionality 263

37.2 Applicability of our attack 264

37.2.1 A concrete example 265

37.2.2 Our attack against other candidate constructions . . . 266

38 A patch for indistinguishability? 267

38.1 A weaker indistinguishability notion 267

38.2 The shortcoming of this definition 268

38.2.1 Identity-based encryption 269

38.2.2 Attribute-based encryption 270

38.2.3 Incompatibility with predicate encryption 272

39 Open Questions 273

Bibliography V 275

Chapter 32

A short review of functional encryption

32.1 Historical motivation

The primary goal of cryptography was always to assure a secure communication channel between two or more parties. The presence of malicious opponents made it necessary to use on the first end an encryption method hiding messages and on the other end a decryption method recovering them. Classically, it was believed that this is the best that can be expected from cryptographic schemes. However, *fully-homomorphic encryption* [Gen09] challenged this traditional view by allowing evaluations over encrypted data. In this setting, a user can perform an operation on a ciphertext in such a way that the decryption of the result corresponds to the initial message to which the same operation has been applied to. As only a holder of the secret key is able to decrypt ciphertexts, computations on the encrypted data can be outsourced to potentially hostile third-parties. Yet, as no information on the evaluations can be obtained without the secret key, some optimization problems, such as retrieving messages with particular properties, cannot be fully outsourced. A new paradigm known as *functional encryption* seems to solve the latter problem. Introduced by *O'Neill*, *Boneh*, *Sahai*, and *Waters* [O'N10, BSW11], functional encryption allows to evaluate encrypted data and obtain the result in plain without leaking disclosed information on the underlying data. To achieve this level of flexibility, a holder of the secret key must provide an evaluator with a functional key that can be used for the desired evaluation. Intuitively, a functional key permits to learn a specific property of the underlying data but does not reveal other information.

32.2 Applications

There exists a wide plethora of applications of functional encryption. Indeed, [GKP+13] illustrates its use for *data-mining* where surgical access over data is crucial. In particular, this includes searching for keywords over encrypted data or encrypted image recognition. [BGI14] extends the definitional landscape to *functional signatures* and *functional pseudorandom functions*. Both notions inherit the properties of their classical counterparts but present some more flexibility. For example, functional signature schemes provide a master signing key that can be used to sign any message and a set of functional signing keys that can only be used to sign a message with particular properties. This may be of medical use where a doctor can prescribe any medicine, but a nurse is restricted to a specific set of antibiotics. Another recent stream of works studies the relationship between functional encryption and *indistinguishability obfuscation* [GGH+13, AJ15, BV18, AFH+20, LPST16, JLS21] which targets to render a program unintelligible but still functional. Besides those novel applications, we should not forget about the classical primitives that are generalized by functional encryption, such as traditional *public-key encryption* [DH76], *identity-based encryption* [Sha85], and *attribute-based encryption* [SW04], each with its own important range of applications.

32.3 Instantiations

Despite its novelty, several candidate functional encryption schemes have already been developed. Some of them were designed to support general circuits of polynomial size [GGH+13, GGHZ16], others to support multi-input functions, where each input can be encrypted independently [LPST16, KS17, JLS21, BKS16]. Both groups include developments in the private-key and public-key setup. Notable schemes, such as [GKP+13] prove the existence of functional encryption based on *fully-homomorphic encryption*, *attribute-based encryption*, and *garbling schemes* [BGG+14]. Last but not least, some realizations of functional encryption concentrate on simple functionalities such as inner-products [ABDCP15, BBL17, BJK15, DDM16, Lin16, BCFG17, CLT18] or quadratic functionalities [BCFG17].

Chapter 33

Functional encryption

Functional encryption emerged in the *public-key setting* [O'N10, BSW11] but has recently been adapted to the *private-key setting* [BKS16, KS17]. Whereas in public-key functional encryption schemes the master public key, needed to encrypt messages, is publicly available, the private-key framework restricts encryption privileges to holders of the master private key. This difference is significant from multiple points of view including efficiency, malleability, and use cases. Hereinafter, we consider the public-key context only and we refer to [BKS16, KS17] for an overview of the private-key setting.

As described in Section 32.1, functional encryption arouse as an ambitious paradigm allowing evaluations over encrypted data. To define a functional encryption scheme, we need to fix which operations shall be supported; in other words, we need to define its *functionality* [O'N10, BSW11, BO13].

Definition 33.1 (Functionality)**.** A functionality $\mathcal{F}$ defined over $(\mathcal{K}, \mathcal{M})$ is a function $\mathcal{F} : \mathcal{K} \times \mathcal{M} \to \Sigma \cup \{\perp\}$ where the set $\mathcal{K}$ is the key space, the set $\mathcal{M}$ is the plaintext message space, the set Σ is the output space, and $\perp$ is a special error symbol not contained in Σ.

Given a specific functionality $\mathcal{F}$, we may devise a corresponding functional encryption scheme. Just like traditional public-key encryption schemes, a functional encryption scheme has a general setup that generates the *master secret key* msk that is kept secret and the *master public key* mpk that is available to everyone. Using the master public key, anyone can encrypt a message m and obtain the corresponding ciphertext CT. Decryption deviates from the classical setting as no one should be able to recover the plaintext m, but only a specific evaluation $\mathcal{F}(k, m)$. To grant permission for such an evaluation, a holder of the master secret key generates a functional key

sk_k corresponding to a key k. This functional key sk_k can then be used to evaluate the ciphertext CT. If someone attempts to evaluate the ciphertext without knowledge of such a functional key, the decryption procedure is allowed to fail and return the special error symbol $\perp$. Formally, we retrieve the following definition [O'N10, BSW11, BO13].

Definition 33.2 (Functional Encryption Scheme - Public-Key Setting)**.** A functional encryption scheme FE in the public-key setting for a functionality $\mathcal{F}$ over $(\mathcal{K}, \mathcal{M})$ consists of a quadruple of algorithms ($\mathsf{FE.Setup}$, $\mathsf{FE.KDer}$, $\mathsf{FE.Enc}$, $\mathsf{FE.Dec}$) such that:

1. $(\mathrm{msk}, \mathrm{mpk}) \leftarrow\!\!\$\ \mathsf{FE.Setup}(\lambda)$: given a security parameter λ, it outputs a pair of master secret/public keys.

2. $\mathsf{CT} \leftarrow\!\!\$\ \mathsf{FE.Enc}(\mathrm{mpk}, m)$: the randomized encryption procedure encrypts the plaintext $m \in \mathcal{M}$ under the master public key mpk.

3. $\mathrm{sk}_k \leftarrow\!\!\$\ \mathsf{FE.KDer}(\mathrm{msk}, k)$: using the master secret key, the (possibly randomized) key-derivation procedure outputs a functional key sk_k corresponding to a key $k \in \mathcal{K}$.

4. $y \leftarrow \mathsf{FE.Dec}(\mathrm{sk}_k, \mathsf{CT})$ decrypts the ciphertext CT using a functional key sk_k to learn either a valid message evaluation $\mathcal{F}(k, m)$ or the special error symbol $\perp$.

We say that a public-key functional encryption scheme FE is correct if for all $m \in \mathcal{M}$ and all $k \in \mathcal{K}$ such that $\mathcal{F}(k, m) \neq \perp$, the following holds:

$$\Pr\left[y = \mathcal{F}(k,m) \;\middle|\; \begin{array}{l} (\mathrm{msk}, \mathrm{mpk}) \leftarrow\!\!\$\ \mathsf{FE.Setup}(\lambda) \wedge \\ \mathsf{CT} \leftarrow\!\!\$\ \mathsf{FE.Enc}(\mathrm{mpk}, m) \wedge \\ \mathrm{sk}_k \leftarrow\!\!\$\ \mathsf{FE.KDer}(\mathrm{msk}, k) \wedge \\ y \leftarrow \mathsf{FE.Dec}(\mathrm{sk}_k, \mathsf{CT}) \end{array} \right] \in 1 - \textsc{Negl}(\lambda) \ .$$

Following [BO13], we stress that correctness makes no requirement on what happens when $\mathcal{F}(k, m) = \perp$. If this is the case, decryption might also result in the error symbol $\perp$, but it is not forced to do so.

Chapter 34

Security notions

The greatest challenge for functional encryption is to define a universal, reliable, and non-restricting security notion. A suitable definition should not only capture the security concepts of the generalized traditional primitives but also foresee precautions for new threats. We rapidly revise a variety of distinct security notions for functional encryption.

34.1 Data-privacy

The first and most important security notion consists in *data-privacy* [O'N10]. Traditionally, data-privacy protects the confidentiality of the encryptor by hiding the encrypted plaintext messages. Functional encryption requires further that no information about the plaintext messages can be gained from the performed evaluations. In symbols, data-privacy implies that the encrypted evaluation of $\mathcal{F}(m, k)$ does not leak any information about the message m and stresses in particular the role of the ciphertext CT. Data-privacy for functional encryption is essentially covered by two concurrent notions: *indistinguishability* [O'N10], claiming that an adversary cannot distinguish between the encryptions of two known plaintext messages, and *semantic security* [O'N10], capturing the idea that an adversary cannot gain more information from a ciphertext than what can be obtained from the functional keys and their corresponding function evaluations. Roughly speaking, indistinguishability directly compares ciphertexts, whereas semantic security relies on the computational indistinguishability of ciphertext distributions. These notions are deepened in Chapter 35.

34.2 Function-hiding

Function-hiding [BRS13], also known as *function-privacy* or *key-hiding*, captures the idea that "it should not be possible to learn any information, beyond the absolute minimum necessary" on the key defining a functional key. It protects the evaluator by hiding the information about the evaluations he desires to carry out. In symbols, function-privacy implies that the encrypted evaluation of $\mathcal{F}(m, k)$ does not leak any information about the queried key k and stresses, in particular, the role of the functional key sk_k. It is crucial to note that the word *function* in this setting makes neither reference to the functionality $\mathcal{F}$, nor to its internal function f (see Section 36.1), but only to the key k. As k can be thought of as defining $\mathcal{F}(m, k)$ (a message m can be evaluated on multiple keys), pioneer works referred to k as "*the function*" of the evaluation.

Despite its utility, [BS15] points out that "in the public-key setting, where anyone can encrypt messages, only a limited form of function-privacy can be satisfied". Yet, slightly restricting the adversarial privileges in the original definition, one may devise a satisfiable counterpart. Two lines of work were devoted to such a construction: [BRS13] formalizes a complementary notion to data-privacy by invoking a new security game that is independent of data-privacy. [AAB$^+$13] devises an umbrella notion that captures simultaneously data-privacy and function-privacy. Both constructions are built on simulation security.

34.3 Consistency and robustness

Consistency [BKKW21] grants protection to the decryptor by preventing the encryptor from crafting ciphertexts that cannot be related to valid messages, even though they are decrypted to some valid function evaluation. It strengthens the correctness of encryption and avoids fraud on valid evaluations. *Robustness* ensures that "a ciphertext cannot correctly decrypt under two different secret keys" [FLPQ13]. It makes encryption schemes more *mis-use resistant* [ABN10] and is a desirable feature for many schemes. We highlight that robustness advocates that if a message is evaluated under two different keys, then at least one evaluation returns the special error symbol $\perp$. However, it does not predict the evaluation behaviour of two distinct messages under the same key.

Chapter 35

Focus on data-privacy

This chapter is devoted to the study of two data-privacy notions for functional encryption, namely *indistinguishability* and *semantic security*. Hereinafter, we let FE be any functional encryption scheme for the functionality $\mathcal{F} : \mathcal{K} \times \mathcal{M} \to \Sigma \cup \{\perp\}$.

35.1 Indistinguishability

Indistinguishability [O'N10] for functional encryption consists in a pragmatic security game attesting data-privacy if it can only be won with probability $\frac{1}{2}$. In the game, the adversary is allowed to choose two messages m_0, m_1 one of which will be encrypted by the challenger. Subsequently, the adversary can ask for functional keys and shall distinguish which message has been encrypted. However, to ensure that the adversary cannot trivially win the game by choosing two keys with distinct evaluations for m_0 and m_1, we need to limit the adversary's querying capacity to keys leading to the same evaluation under m_0 and m_1. Formally, we deduce the following definition [BSW11, BO13].

Definition 35.1. We say that a public-key functional encryption scheme FE is indistinguishably secure against chosen plaintext attacks ($\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-secure) if the advantage of any PPT adversary $\mathcal{A}$ against the $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-game defined in Figure 35.1 is negligible:

$$\mathbf{Adv}^{\mathsf{IND-FE-CPA}}_{\mathcal{A},\mathsf{FE}}(\lambda) := \left| \Pr\left[\mathsf{IND} - \mathsf{FE} - \mathsf{CPA}^{\mathcal{A}}_{\mathsf{FE}}(\lambda) = 1\right] - \frac{1}{2} \right| \in \textsc{Negl}(\lambda).$$

Indistinguishability, is subdivided into *selective* and *adaptive* security. In the selective security game ($s\mathsf{IND} - \mathsf{FE} - \mathsf{CPA}$), the adversary starts the game

```
$s\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}^{\mathcal{A}}_{\mathsf{FE}}(\lambda)$:
  $b \leftarrow_\$ \{0,1\}$
  $\mathrm{L} \leftarrow \emptyset$
  $(m_0, m_1) \leftarrow_\$ \mathcal{A}(\lambda)$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow_\$ \mathsf{FE.Setup}(\lambda)$
  $\mathsf{CT} \leftarrow_\$ \mathsf{FE.Enc}(\mathrm{mpk}, m_b)$
  $b' \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) \neq \mathcal{F}(k, m_1)$
      return 0
  return $b \stackrel{?}{=} b'$

Proc. $\mathrm{KDerO}_{\mathrm{msk}}(k)$:
  $\mathrm{L} \leftarrow \mathrm{L} \cup \{k\}$
  $\mathrm{sk}_k \leftarrow_\$ \mathsf{FE.KDer}(\mathrm{msk}, k)$
  return $\mathrm{sk}_k$
```

```
$a\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}^{\mathcal{A}}_{\mathsf{FE}}(\lambda)$:
  $b \leftarrow_\$ \{0,1\}$
  $\mathrm{L} \leftarrow \emptyset$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow_\$ \mathsf{FE.Setup}(\lambda)$
  $(m_0, m_1) \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathrm{mpk})$
  $\mathsf{CT} \leftarrow_\$ \mathsf{FE.Enc}(\mathrm{mpk}, m_b)$
  $b' \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) \neq \mathcal{F}(k, m_1)$
      return 0
  return $b \stackrel{?}{=} b'$

Proc. $\mathrm{KDerO}_{\mathrm{msk}}(k)$:
  $\mathrm{L} \leftarrow \mathrm{L} \cup \{k\}$
  $\mathrm{sk}_k \leftarrow_\$ \mathsf{FE.KDer}(\mathrm{msk}, k)$
  return $\mathrm{sk}_k$
```

Figure 35.1: The indistinguishability chosen-plaintext experiments defined for a public-key functional encryption scheme. In both games, the adversary $\mathcal{A}$ can use the key derivation procedure $\mathrm{KDerO}_{\mathrm{msk}}$ as often as desired.

by choosing the plaintext messages m_0, m_1. Subsequently, the functional encryption setup takes place. In this scenario, functional key queries are only allowed after the plaintexts have been chosen. In the adaptive security game ($a\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$), the functional encryption setup takes place before the adversary needs to choose the plaintext messages m_0, m_1. In this case, the adversary may query for functional keys before choosing the plaintext messages. This allows him to adapt his plaintext choice to the received functional keys. In both cases, the early loss condition for forbidden functional key queries holds.

Remark 35.2. *We note that adaptive security implies selective security, but the reciprocal does not necessarily hold. In this sense, selective security is a weaker security notion.*

Unfortunately, neither selective nor adaptive security perfectly reflects our intuition on security as they fail to capture some elementary counterexamples. [BSW11] investigates the case of the trivial functionality defined by $\mathcal{F}(k, m) = m$ for all $k \in \mathcal{K}$. For this functionality, the intuitively insecure encryption scheme $\mathsf{Enc}(\mathrm{mpk}, m) = m$ passes the indistinguishability game. Indeed, due to the forbidden query condition, an adversary is only allowed to issue plaintext messages $m_0 = m_1$ which are trivially indistinguishable. Thus, the adversary cannot win the game. This loophole is a severe drawback, as the trivial functionality $\mathcal{F}(k, m) = m$ is used to recover the traditional public-key paradigm: the functionality and the key derivation procedure have no effect, such that only the encryption and decryption procedures are applied.

35.2 Semantic security

Semantic security captures the idea that a secret key sk_k should only reveal the function evaluation $\mathcal{F}(k, m)$ and no other information. It grants security under so-called key-revealing selective opening attacks (**SS1**). Unfortunately, such a strong security notion seems to be unachievable: [BSW11] proved this claim in the non-programmable random oracle model and [BO13] in the standard model. To replace this utopic semantic security notion, [BO13] proposes two new definitions **SS2** and **SS3**. **SS2** is equivalent to $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-security for all functionalities and experiences the same drawbacks. **SS3** seems to finally capture an achievable and superior security notion. Indeed, contrary to indistinguishability, **SS3**-security successfully rules out the trivial encryption scheme $\mathsf{Enc}(\mathrm{mpk}, m) = m$ from being secure because it detects message leakage through publicly available information.

35.3 Informal comparison

[BO13] shows that **SS3** is strictly stronger than $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$. This means that any **SS3**-secure functional encryption scheme is also $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-secure, but the reciprocal might not hold, as illustrates the trivial encryption scheme $\mathsf{Enc}(\mathrm{mpk}, m) = m$. Yet, this notion of semantic security coincides with the indistinguishability notion for a large class of functionalities. The disadvantage of **SS3** is its dependence on the indistinguishability of ciphertext distributions, which is hard to handle in practice. $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$ provides a pragmatic advantage as the indistinguishability of ciphertexts is sufficient to testify security. We note the difference between ciphertext distributions and individual ciphertexts. In the latter case, the cumulative role of the underlying distributions is not taken into account. Although slightly weaker, $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$ continues to be used in practice. Hereinafter, we focus on indistinguishability only, as any scheme that is not $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-secure cannot be **SS3**-secure. Concretely, we devise a conditional attack against indistinguishability which has an impact on a range of DDH-based inner-product functional encryption schemes when restricting them to bounded-norm evaluations.

Chapter 36

Our attack

In this chapter, we provide a selective chosen-plaintext attack which shows that under some conditions, functional encryption schemes cannot achieve indistinguishability. We start by decomposing a functionality into a mathematical function and an error trigger. This separation raises the question of where such an error trigger may appear in a functional encryption scheme. Upon discussing the position of the error trigger, we revise the indistinguishability notion and point out a potential threat of functional encryption schemes which leaves the scope of traditional public-key encryption. Finally, we devise a conditional attack against error-proned functional encryption schemes.

We highlight that our development crucially relies on the special role of the error symbol $\perp$ in functional encryption. Therefore, our observations apply to error-proned constructions only. Error-free schemes, such as [ALS16] and other post-quantum lattice-based constructions, for which $\mathcal{F}(k,m) \neq \perp$ for all $m \in \mathcal{M}$ and $k \in \mathcal{K}$ do not underlie our attack.

Remark 36.1. *The central element of the upcoming analysis is the separation of a functionality into an internal function and an error trigger. This allows us to study the relation between error evaluations and indistinguishability. We point out that there is an analogous development in [BKKW21] with respect to function-hiding. In that work, the authors assigned different error symbols to error evaluations, each depending on a distinct error cause. The resulting consistency attack in [BKKW21, Section 5.B] is similar to our indistinguishability attack but focusses on keys instead of messages.*

36.1 Functionalities and their error symbols

A functionality $\mathcal{F} : \mathcal{K} \times \mathcal{M} \to \Sigma \cup \{\perp\}$ is a function that is allowed to fail and return a special error symbol $\perp$. Albeit the error symbol seems to be meaningless, it is required to successfully capture a family of cryptographic primitives (see Section 38.2). Abstractly, a functionality may be decomposed into an internal function $f : \mathcal{K} \times \mathcal{M} \to \Sigma'$ with $\Sigma \subseteq \Sigma'$ and an error trigger $\mathcal{E} : \Sigma' \to \Sigma \cup \{\perp\}$ such that $\mathcal{E}(y) = y$ for all $y \in \Sigma$ and $\mathcal{E}(y) =\perp$ otherwise (see [DS00, KLZ96] for the general concept of decompositions). In this interpretation, f can be preceded by yet another error trigger ruling out incompatible function inputs. This auxiliary error trigger should be thought of as returning a different error symbol and may not be needed if only valid inputs are given to the functionality. Hereinafter, we restrict the inputs of the functionality to $\mathcal{K}$ and $\mathcal{M}$ so that this auxiliary error trigger is not needed.

36.2 Functional encryption and error triggers

A functional encryption scheme for the functionality $\mathcal{F} : \mathcal{K} \times \mathcal{M} \to \Sigma \cup \{\perp\}$ mimics the functionality $\mathcal{F}$ but hides the plaintext message space $\mathcal{M}$. In practice, many functional encryption schemes rely on the decomposed functionality described in Section 36.1. Indeed, a suitable internal function and an error trigger can be directly deduced from the functionality. If the corresponding construction is not used in a black-box manner, this decomposition is public. Before we continue to investigate the impact of this publicly available information, we discuss which one of the four functional encryption sub-algorithms FE.Setup, FE.KDer, FE.Enc, FE.Dec allows the implementation of the error trigger:

FE.Gen: The key generation algorithm generates the master secret key and master public key for the functional encryption scheme. It is not related to functional evaluations and therefore the error trigger would have no effect.

FE.Enc: The encryption algorithm encrypts admissible plaintext messages with respect to the master public key. It is not associated with the functional keys, so the error trigger would miss one input value. Undesired input messages can be avoided by shrinking the plaintext message space $\mathcal{M}$.

FE.KDer: The key derivation algorithm generates functional keys. It is not associated with ciphertexts and so the error trigger would miss one input value. Undesired key queries can be avoided by shrinking the query space $\mathcal{K}$.

FE.Dec: The decryption algorithm decrypts a given ciphertext using a suitable functional key to learn the outcome of the functional evaluation of a message and a queried key. The decryption algorithm consists in the only procedure that has simultaneous access to plaintext message and key information. Thereby, it is the only sub-algorithm that can contain the error trigger.

Thus, an error trigger may only be implemented in the decryption procedure of a functional encryption scheme. If we stick to the original error trigger $\mathcal{E}$, then an input from Σ' is required to capture the intuition that some information about $f(k, m)$ is processed. However, such a direct input may not be desired. In this case, the original error trigger $\mathcal{E}$ may be replaced by an equivalent one $\mathcal{E}' : \Omega \to \Sigma \cup \{\perp\}$ with another input set Ω. Nonetheless, the error trigger $\mathcal{E}'$ needs to process the information $f(k, m)$.

36.3 Revision of indistinguishability

A functional encryption scheme is $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-secure if an adversary can only win the indistinguishability game in Figure 35.1 with negligible advantage. Our focus lies on the weakest indistinguishability notion, namely selective security against chosen-plaintext attacks ($s\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-security). In this game, an adversary first chooses two admissible plaintext messages m_0 and m_1. After handing those messages to the challenger, the functional encryption scheme is set up. The functioning of the functional encryption scheme is known to the adversary, but he ignores the randomly chosen parameters. Upon the generation of the master secret key and the master public key, a message will be encrypted at random. The adversary shall now distinguish which message has been chosen. For this task, he has oracle access to the key derivation procedure and may use the ciphertext and the master public key. Furthermore, the decryption procedure is public and can be used to receive a function evaluation of a queried key and the challenge message. To ensure that the adversary cannot trivially win the game by requesting the functional key for some key k such that $\mathcal{F}(k, m_0) \neq \mathcal{F}(k, m_1)$, $\mathsf{IND} - \mathsf{FE} - \mathsf{CPA}$-security forbids such queries by letting the adversary lose the game.

Although compromising key queries are foreseen in the indistinguishability notion, it is falling short to tackle another practical problem. If a functional encryption scheme contains an error trigger in its decryption procedure, then the decryption procedure also assembles some information $f(k, m)$ of the internal function of the functionality. An adversary may be able to simulate the decryption procedure and partially filter out this information.

36.4 Attacking indistinguishability

Assume first that the decryption procedure computes the value $f(k, m)$ and subsequently applies $\mathcal{E}(f(m, k))$ to reveal $\mathcal{F}(k, m)$. Then, the adversary can exploit a loophole in the $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-security definition to always win its security game. More precisely, choosing two plaintext messages m_0, m_1, and a key k such that $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1) =\perp$ but $f(k, m_0) \neq f(k, m_1)$, the adversary can always win the game. As $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1) =\perp$, the adversary does not loose the game through a forbidden functional key query. As m_0, m_1, k are known, the adversary may compute the two distinct values $f(k, m_0), f(k, m_1)$. By simulating the decryption procedure with the challenge ciphertext and the functional key sk_k, but stopping before triggering an error, he recovers exactly one of the values $f(k, m_0), f(k, m_1)$ and finds out which message has been encrypted.

Of course, no candidate functional encryption scheme involves such a plain intermediate value. Generally, the intermediate values are obfuscated through secure cryptographic primitives such as *one-way functions*. Unfortunately, if the underlying one-way function is known, then the same trick still compromises $\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-security.

Theorem 36.2. *Let $\mathcal{F} : \mathcal{K} \times \mathcal{M} \to \Sigma \cup \{\perp\}$ be a functionality, let $f : \mathcal{K} \times \mathcal{M} \to \Sigma'$ be its internal function where $\Sigma \subseteq \Sigma'$, and let $\mathcal{E} : \Sigma' \to \Sigma \cup \{\perp\}$ be its error trigger such that $\mathcal{F} = \mathcal{E} \circ f$ where $\mathcal{E}(y) = y$ for all $y \in \Sigma$ and $\mathcal{E}(y) =\perp$ otherwise. Let FE be a functional encryption scheme for the functionality $\mathcal{F}$. Let $\mathbf{OWF} : \Sigma' \to \mathbb{G}$ be a known efficiently computable one-way function used in the functional encryption scheme FE. Assume that for a key $k \in \mathcal{K}$ and a plaintext message $m \in \mathcal{M}$, the decryption procedure computes $\mathbf{OWF}(f(k, m))$ as an intermediate value. If there exist $k \in \mathcal{K}$ and $m_0, m_1 \in \mathcal{M}$ such that $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1) =\perp$ but $\mathbf{OWF}(f(k, m_0)) \neq \mathbf{OWF}(f(k, m_1))$, then the functional encryption scheme FE is not $s\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-secure.*

Proof. We are going to play the $s\mathsf{IND} - \mathsf{FE} - \mathsf{CPA}$-security game from Figure 35.1 against the functional encryption scheme FE. We impersonate an adversary. We start the game by releasing the plaintext messages m_0 and m_1. Then, $\mathsf{FE.Setup}$ generates the master secret key msk and the master public key mpk. Next, one of our plaintext messages m_b with $b \in \{0, 1\}$ is encrypted by $\mathsf{FE.Enc}$ using the master public key and we obtain the corresponding ciphertext CT. Subsequently, we use the $\mathsf{FE.KDer}$ oracle once to query for the functional key sk_k corresponding to k. This query is valid as $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1) =\perp$, in other words, we do not lose the game due to a forbidden functional key query. After this, we simulate the decryption procedure $\mathsf{FE.Dec}$. We run through all the intermediate decryption steps but stop when we obtain $g_b = \mathbf{OWF}(f(k, m_b))$ (i.e., before recovering $\mathcal{F}(k, m_b)$). Once we obtain g_b, we compute $g_0 = \mathbf{OWF}(f(k, m_0))$ and $g_1 = \mathbf{OWF}(f(k, m_1))$. This is possible as we know k, m_0, m_1 in plain, as well as the internal function f and the used one-way function $\mathbf{OWF}$. As $\mathbf{OWF}(f(k, m_0)) \neq \mathbf{OWF}(f(k, m_1))$, either $g_0 = g_b$ or $g_1 = g_b$, but not both. Thus, we can distinguish with certainty which of our initial messages has been encrypted and win so the security game. □

Theorem 36.2 is independent of the chosen error trigger, whose exact functioning may be difficult to simulate. It only uses the internal function f that must be hard-coded into the scheme. Roughly speaking, our attack only requires that:

1. The decryption procedure computes $\mathbf{OWF}(f(k, m))$, where $\mathbf{OWF}$ and f are known, and
2. there exist $k \in \mathcal{K}$ and $m_0, m_1 \in \mathcal{M}$ such that:
   (a) $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1) =\perp$, but
   (b) $\mathbf{OWF}(f(k, m_0)) \neq \mathbf{OWF}(f(k, m_1))$.

It is clear that any scheme that successfully avoids error evaluations such that $\mathcal{F}(k, m) \neq\perp$ for all $k \in \mathcal{K}$ and $m \in \mathcal{M}$, does not fulfil condition 2.(a) above and is not impacted by our attack. Such a scheme may be achieved by restricting inputs as described in [ACF+17] and [ABDCP15, Construction 4.1] or by using an error-free decryption algorithm [ALS16].

Remark 36.3. *A similar attack applies if* $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1) = s \in \Sigma$ *and* $\mathbf{OWF}(f(k, m_0)) \neq \mathbf{OWF}(f(k, m_1))$*, but usually* $f(k, m_0) = f(k, m_1)$ *in this case.*

36.5 Summary

The pragmatic decomposition of a functionality $\mathcal{F} : \mathcal{K} \times \mathcal{M} \to \Sigma \cup \{\perp\}$ into an internal function $f : \mathcal{K} \times \mathcal{M} \to \Sigma'$ with $\Sigma \subseteq \Sigma'$ and an error trigger $\mathcal{E} : \Sigma' \to \Sigma \cup \{\perp\}$ such that $\mathcal{E}(y) = y$ for all $y \in \Sigma$ and $\mathcal{E}(y) = \perp$ otherwise, in combination with the special status of the error symbol allows for a conditional attack against the security of functional encryption schemes. Any scheme underlying this attack is not $s\mathsf{IND} - \mathsf{FE} - \mathsf{CPA}$ secure which consists in the weakest data-privacy notion for functional encryption. In particular, Theorem 36.2 shows that it is not sufficient to hide the internal function f through the means of a one-way function as it can be reverse-engineered. Hence, Theorem 36.2 simultaneously yields a necessary criterion for $s\mathsf{IND} - \mathsf{FE} - \mathsf{CPA}$ and provides a pragmatic sanity test. Furthermore, it highlights that the design of the decryption procedure requires special attention.

Chapter 37

Impact on the bounded-norm inner-product functionality

To further motivate our attack, we show how it compromises a family of candidate functional encryption schemes for the *inner-product functionality* when restricting them to bounded-norm evaluations.

37.1 Bounded-norm inner-product functionality

The bounded-norm inner-product functionality essentially multiplies two vectors and checks if the result is sufficiently small. Abstractly, we get the following definition.

Definition 37.1 (Bounded-norm inner-product functionality)**.** Let $(\mathbf{y}, \mathbf{x}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$ where $\mathbb{Z}_q = \mathbb{Z} \cap \left(-\frac{q}{2}, \frac{q}{2}\right]$ and let $B < q$. We define $IP_B(\mathbf{y}, \mathbf{x}) := [\mathbf{x} \cdot \mathbf{y}^\top \mod q]$, if $|[\mathbf{x} \cdot \mathbf{y}^\top \mod q]| \leq B$ and $IP_B(\mathbf{y}, \mathbf{x}) := \perp$ otherwise.

Remark 37.2. *In the literature, the inner-product functionality is not restricted to bounded-norm evaluations, however, in practice, it is often used in this way, for example when discrete logarithms need to be computed. Choosing q and B sufficiently large and keeping the entries of $\mathbf{x}$ and $\mathbf{y}$ sufficiently small results in the traditional non-restricted inner-product functionality.*

Despite its simplicity, a functional encryption scheme for the inner-product functionality may find attractive applications. [ABDCP15] gives the example of weighted averages that need to be computed without leaking any information on the message vector. Another motivational example is the outsourcing of tax-related computations:

Assume that a national fiscal agency stores the monthly incomes of its citizens in an encrypted format. The data of a citizen can be thought of as a vector $\mathbf{x} = (x_1, \ldots, x_{12})$ representing his monthly income. To increase the annual budget, the ministry of financial affairs wants to introduce a new differential tax scale that reduces taxes at the beginning of the year, but increases them by the end. In this way, they would gain more from the interests of the citizens' bank deposits that accumulate over the whole year. For example, a potential tax scale could look like 1% for January, 2% for February, ..., 12% for December and can be represented by a vector $\mathbf{y} = (1, \ldots, 12)$. To determine a tax scale that achieves the highest income but has the lowest impact on the household budget of its citizens, the government needs to carry out a variety of simulations. Unfortunately, the national fiscal agency does not have the computing power to carry out such a large-scale simulation. Thus, the government decides to outsource these computations to an audit company. The functional encryption scheme allows the audit company to query the database and perform the simulation without knowing the actual financial assets of the citizens. In this scenario, q needs to be chosen sufficiently large to avoid overflows, and B may be set such that the impact on high-income households is discarded from the results.

37.2 Applicability of our attack

Considering the bounded-norm inner-product functionality, we recognize its internal function $f : \mathbb{Z}_q^n \times \mathbb{Z}_q^n \to \mathbb{Z}_q$ to be defined by $f(\mathbf{y}, \mathbf{x}) := \left[\mathbf{x} \cdot \mathbf{y}^\top \mod q\right]$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^n$ and the corresponding error trigger to be defined by $\mathcal{E} : \mathbb{Z}_q \to \mathbb{Z}_q \cup \{\perp\}$ where $\mathcal{E}(k) = k$ whenever $|k| < B$ and $\perp$ otherwise. If B is sufficiently small (e.g. $0 < B < \frac{q-1}{2}$), there are multiple evaluation pairs $(\mathbf{y}, \mathbf{x})$ that evaluate to the error symbol. For example $\mathbf{x}_0^* = (B+1, 1, \ldots, 1)$, $\mathbf{x}_1^* = (-B-1, 1, \ldots, 1)$, and $\mathbf{y}^* = (1, 0, \ldots, 0)$. Thus, if a bounded-norm inner-product functional encryption scheme uses a known one-way function $\mathbf{OWF}$ to hide intermediate values $f(\mathbf{y}, \mathbf{x})$, and if for at least two evaluation pairs $(\mathbf{y}, \mathbf{x}_0)$, $(\mathbf{y}, \mathbf{x}_1)$, we have $\mathbf{OWF}(f(\mathbf{y}, \mathbf{x}_0)) \neq \mathbf{OWF}(f(\mathbf{y}, \mathbf{x}_1))$, then our attack from Theorem 36.2 applies.

Remark 37.3. *The upcoming schemes were designed for the traditional inner-product functionality, which is not restricted to bounded-norm evaluations. Thereby, our attack does not directly contradict their security proofs. However, the need of these schemes to compute discrete logarithms makes the discussion on bounded-norm evaluations relevant.*

37.2.1 A concrete example

In [ABDCP15, Construction 3.1], the authors propose an elementary inner-product functional encryption scheme:

1. The master secret key (msk) is a uniformly at random sampled vector $\mathbf{s} \in \mathbb{Z}_q^n$, and the master public key (mpk) is $g^{\mathbf{s}} := (g^{\mathbf{s}_1}, \dots, g^{\mathbf{s}_n})$ where $g \in \mathbb{G}$ is a generator of the multiplicative group $\mathbb{G}$.

2. To encrypt a vector $\mathbf{x} \in \mathbb{Z}_q^n$, first $r \in \mathbb{Z}_q$ is sampled uniformly at random and then the ciphertext is computed as
$$\mathsf{CT} := (g^r, g^{r\mathbf{s}+\mathbf{x}}) := (g^r, (g^{\mathbf{s}_1})^r \cdot g^{\mathbf{x}_1}, \dots, (g^{\mathbf{s}_n})^r \cdot g^{\mathbf{x}_n}).$$

3. The functional key corresponding to a vector $\mathbf{y} \in \mathbb{Z}_q^n$ is generated as $\mathrm{sk}_{\mathbf{y}} := \left[\mathbf{s} \cdot \mathbf{y}^\top \mod q\right]$.

4. Decryption is carried out through a look-up table containing g^b for all $|b| < B$. If $\left(\prod_{i=1}^n \mathsf{CT}_i^{\mathbf{y}_i}\right) \cdot \left(\mathsf{CT}_0^{\mathrm{sk}_{\mathbf{y}}}\right)^{-1}$ in $\mathbb{G}$, corresponding to $g^{\left[\mathbf{x}\cdot\mathbf{y}^\top \mod q\right]}$, is found inside the look-up table, the corresponding discrete logarithm is obtained, otherwise an error is thrown. In other words, $IP_B(\mathbf{x}, \mathbf{y}) = \left[\mathbf{x} \cdot \mathbf{y}^\top \mod q\right]$ if the value $g^{\left[\mathbf{x}\cdot\mathbf{y}^\top \mod q\right]}$ was in the lookup table and $\perp$ otherwise.

This scheme makes use of the one-way function $\mathbf{OWF} : \mathbb{Z}_q \to \mathbb{G}$ sending an element $t \in \mathbb{Z}_q$ to its cyclic group encoding $g^t \in \mathbb{G}$. Furthermore, the decryption procedure computes $\mathbf{OWF}(f(\mathbf{y}, \mathbf{x})) = g^{\left[\mathbf{x}\cdot\mathbf{y}^\top \mod q\right]}$ as an intermediate value. As for $\mathbf{x}_0^* = (B+1, 1, \dots, 1)$, $\mathbf{x}_1^* = (-B-1, 1, \dots, 1)$ and $\mathbf{y}^* = (1, 0, \dots, 0)$, we have $\mathbf{OWF}(f(\mathbf{y}^*, \mathbf{x}_0^*)) = g^{B+1}$, and $\mathbf{OWF}(f(\mathbf{y}^*, \mathbf{x}_1^*)) = g^{-(B+1)}$. Those values are usually non-equal. Indeed, q is generally an odd prime and $|\mathbb{G}| = q-1$, so that the equality of both values implies that $2(B+1)$ divides a multiple of $q-1$ which is impossible if B is sufficiently small. In the unlikely event that they would be equal, we may choose $\mathbf{x}_1^* = (B+2, 1, \dots, 1)$ which then produces a different value. Thus, all of the conditions of our attack are satisfied and Theorem 36.2 claims that the considered FE scheme is not $s\mathsf{IND}-\mathsf{FE}-\mathsf{CPA}$-secure. For illustration, playing the indistinguishability game with the suggested challenge plaintexts and the corresponding key shows that $IP_B(\mathbf{x}_0^*, \mathbf{y}) = IP_B(\mathbf{x}_1^*, \mathbf{y}) = \perp$ which makes the queries admissible, and regularly computing $\left(\prod_{i=1}^n \mathsf{CT}_i^{\mathbf{y}_i^*}\right) / \mathsf{CT}_0^{\mathrm{sk}_{\mathbf{y}^*}} = g^{\left[\mathbf{x}_b^*\cdot(\mathbf{y}^*)^\top \mod q\right]}$ produces either g^{B+1} or g^{-B-1} which can be distinguished.

37.2.2 Our attack against other candidate constructions

Albeit the scheme from [ABDCP15, Construction 3.1] has been chosen to illustrate our attack, it does not consist in an isolated example. Any inner-product functional encryption scheme based on the DDH assumption using a polynomially bounded look-up table for decryption falls prey to the same attack. The reason is that they all use the same internal function and the same one-way function. Thus, the same arguments can be used to conclude their vulnerability when only bounded-norm evaluations are considered. Other such schemes are [BJK15, DDM16, CLT18].

A clever restriction of the bounded-norm inner-product functionality allowing only short vectors as functionality inputs (i.e., $\max(\mathbf{x}) < X$ and $\max(\mathbf{y}) < Y$ such that $nXY < B$) as described in [ABDCP15, Construction 4.1] or [ACF$^+$17] may prevent error decryption and thus also our attack. Unfortunately, those bounds shrink the number of possible message-key evaluations. Furthermore, no boundary cases, such as a single large entry or mutually eliminating entries, are allowed, which might be hindering for some applications. Additionally, this solution is devised to work over the integers and not over finite integer rings. As modular reductions are disabled its application range shrinks. If simultaneously restricted vector entries are considered, but modular reductions are allowed, like in [Tom19, Section 4.1], then our attack applies again.

The security proofs of the referenced constructions certifying indistinguishability do not consider the bounded-norm condition. Indeed, the proofs consider the non-restricted inner-product functionality and carry out a standard reduction to the DDH assumption. Whereas this reduction is perfectly binding if only valid decryptions exist, the same does not hold true if errors can be obtained, which is the case for the bounded-norm inner-product functionality. Our attack illustrates this auxiliary threat which leaves the scope of traditional public-key encryption. In particular, the simple obfuscation through DDH-encodings is not sufficient to grant indistinguishability. This observation remained unperceived until now.

Nonetheless, there are candidate constructions for the bounded-norm inner-product functionality, such as [ALS16], which do not underlie our attack and do not require any entry restrictions on the vector entries. Such post-quantum constructions give the proof-of-concept that error-free functional encryption schemes are achievable and that the special role of the error symbol may not be needed for this functionality. At the same time, our attack backs them up by proving their superiority compared to error-proned schemes.

Chapter 38

A patch for indistinguishability?

DDH-based constructions are generally pragmatic and easy to implement. As such, they enjoy great interest among practitioners. To complement our attack, we would like to investigate the impact of a slight definitional change weakening indistinguishability but declaring the referenced DDH-based constructions secure for the bounded-norm inner-product functionality.

38.1 A weaker indistinguishability notion

The decisive element of our attack lies in the observation that the functional evaluation of two message-key pairs may be equal but that the underlying internal function evaluated in the same pairs is not. One solution could be to relate the security definition to the internal function instead of the functionality. However, such a definition increases the risk of function-specific attacks and shrinks the number of potential functionalities. From another perspective, we observe that the problematic situation seems only to arise if the functionality evaluates to the error symbol. This can be detected in the output layer of the scheme. Thereby, a pragmatic solution to declare the DDH-based constructions from Section 37.2.2 indistinguishably secure for the bounded-norm inner-product functionality may consist in not only forbidding functional key queries that result in different evaluation values, but also queries whose evaluations return the error symbol. In Figure 38.1 we formalize a corresponding security game.

```
$s\mathsf{IND-FE-CPA}^{\mathcal{A}}_{\mathsf{FE}}(\lambda)$:
  $b \leftarrow\!\!\$ \{0,1\}$
  $\mathrm{L} \leftarrow \emptyset$
  $(m_0, m_1) \leftarrow\!\!\$ \mathcal{A}(\lambda)$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow\!\!\$ \mathsf{FE.Setup}(\lambda)$
  $\mathsf{CT} \leftarrow\!\!\$ \mathsf{FE.Enc}(\mathrm{mpk}, m_b)$
  $b' \leftarrow\!\!\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) \neq \mathcal{F}(k, m_1)$
      return 0
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) = \perp$
      return 0
  return $b \stackrel{?}{=} b'$

Proc. $\mathrm{KDerO}_{\mathrm{msk}}(k)$:
  $\mathrm{L} \leftarrow \mathrm{L} \cup \{k\}$
  $\mathrm{sk}_k \leftarrow\!\!\$ \mathsf{FE.KDer}(\mathrm{msk}, k)$
  return $\mathrm{sk}_k$
```

```
$a\mathsf{IND-FE-CPA}^{\mathcal{A}}_{\mathsf{FE}}(\lambda)$:
  $b \leftarrow\!\!\$ \{0,1\}$
  $\mathrm{L} \leftarrow \emptyset$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow\!\!\$ \mathsf{FE.Setup}(\lambda)$
  $(m_0, m_1) \leftarrow\!\!\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathrm{mpk})$
  $\mathsf{CT} \leftarrow\!\!\$ \mathsf{FE.Enc}(\mathrm{mpk}, m_b)$
  $b' \leftarrow\!\!\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) \neq \mathcal{F}(k, m_1)$
      return 0
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) = \perp$
      return 0
  return $b \stackrel{?}{=} b'$

Proc. $\mathrm{KDerO}_{\mathrm{msk}}(k)$:
  $\mathrm{L} \leftarrow \mathrm{L} \cup \{k\}$
  $\mathrm{sk}_k \leftarrow\!\!\$ \mathsf{FE.KDer}(\mathrm{msk}, k)$
  return $\mathrm{sk}_k$
```

Figure 38.1: Weaker indistinguishability chosen-plaintext experiments defined for a public-key functional encryption scheme. Note that the first early abort check implies that $\mathcal{F}(k, m_0) = \mathcal{F}(k, m_1)$ for all $k \in \mathrm{L}$, and so the second early abort check guarantees that $\mathcal{F}(k, m_0) \neq \perp$ and $\mathcal{F}(k, m_1) \neq \perp$.

38.2 The shortcoming of this definition

Despite introducing a weaker data-privacy notion for functional encryption, the pragmatic patch of Figure 38.1 suffers from another major inconvenience: it is incompatible with a family of classical primitives known as predicate encryption schemes. A *predicate encryption scheme* [KSW08] is a specific form of public-key encryption where secret keys correspond to predicates and ciphertexts are associated with attributes. It generalizes traditional primitives such as *identity-based encryption* [Sha85] and *attribute-based encryption* [GKP+13]. Such primitives can be expressed as particular functionalities [BSW11] and so they may be seen as functional encryption schemes.

Definition 38.1 (Predicate Encryption Functionality)**.** Denote by $\mathcal{P} : \mathcal{K}_0 \times \Omega \rightarrow \{0,1\}$ a polynomial time predicate where $\mathcal{K}_0$ denotes the key space and Ω denotes the attribute space. The Predicate Encryption functionality (for $\mathcal{P}$) is defined by $PEF : \mathcal{K}_0 \times (\Omega \times \mathcal{M}_0) \rightarrow \mathcal{M}_0 \cup \{\perp\}$ such that $PEF(k, (x, m)) = m$ whenever $\mathcal{P}(k, x) = 1$ (i.e., the key and attributes are matching) and $PEF(k, (x, m)) = \perp$ otherwise.

Contrary to bounded-norm inner-product schemes, the security of predicate encryption schemes requires the use of the error symbol. We explain this relation in further detail below.

38.2.1 Identity-based encryption

In *identity-based encryption*, ciphertexts and keys are associated with identities, and a key can only be used to decrypt a ciphertext if the identities match [Sha85]. The upcoming definition stems from [DG21].

Definition 38.2 (Identity-Based Encryption)**.** An identity-based encryption scheme (IBE) for an identity space Ω and an associated message space $\mathcal{M}_0$ is a quadruple of algorithms ($\mathsf{IBE.Setup}$, $\mathsf{IBE.KDer}$, $\mathsf{IBE.Enc}$, $\mathsf{IBE.Dec}$) such that:

1. $(\mathrm{msk}, \mathrm{mpk}) \leftarrow_{\$} \mathsf{IBE.Setup}(\lambda)$: given a security parameter λ, it outputs a pair of master secret/public keys.

2. $\mathsf{CT} \leftarrow_{\$} \mathsf{IBE.Enc}(\mathrm{mpk}, x, m)$: the randomized encryption procedure encrypts the plaintext $m \in \mathcal{M}_0$ with respect to an identity $x \in \Omega$ and the master public key mpk.

3. $\mathrm{sk}_x \leftarrow_{\$} \mathsf{IBE.KDer}(\mathrm{msk}, x)$: using the master secret key msk, the (possibly randomized) key-derivation procedure outputs a functional key sk_x corresponding to the identity $x \in \Omega$.

4. $y \leftarrow \mathsf{IBE.Dec}(\mathrm{sk}_x, \mathsf{CT})$ decrypts the ciphertext CT using the functional key sk_x in order to either learn a message $m \in \mathcal{M}_0$ or the special error symbol $\perp$.

An IBE scheme is correct if for all messages $m \in \mathcal{M}_0$ and all identities $x \in \Omega$, the following holds:

$$\Pr\left[\; y = m \;\middle|\; \begin{array}{l} (\mathrm{msk}, \mathrm{mpk}) \leftarrow_{\$} \mathsf{IBE.Setup}(\lambda) \wedge \\ \mathsf{CT} \leftarrow_{\$} \mathsf{IBE.Enc}(\mathrm{mpk}, x, m) \wedge \\ \mathrm{sk}_x \leftarrow_{\$} \mathsf{IBE.KDer}(\mathrm{msk}, x) \wedge \\ y \leftarrow \mathsf{IBE.Dec}(\mathrm{sk}_x, \mathsf{CT}) \end{array} \right] \in 1 - \textsc{Negl}(\lambda)\ .$$

Identity-based encryption is a special case of predicate encryption where the predicate key space is $\mathcal{K}_0 = \{0, 1\}^*$, the set of attributes is $\Omega = \{0, 1\}^*$

and the predicate is defined as simple comparison (i.e. $x = x^*$). Thus, we may compare its traditional indistinguishability game [DG21] in Figure 38.2 with the original one for functional encryption.

```
$\mathsf{IND} - \mathsf{IBE} - \mathsf{CPA}^{\mathcal{A}}_{\mathsf{IBE}}(\lambda)$:
  $b \leftarrow_\$ \{0,1\}$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow_\$ \mathsf{IBE.Setup}(\lambda)$
  $(x^*, m_0, m_1) \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathrm{mpk})$
  $\mathsf{CT} \leftarrow_\$ \mathsf{IBE.Enc}(\mathrm{mpk}, x^*, m_b)$
  $b' \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $\exists x \in \mathrm{L}$ s.t. $x = x^*$
      return 0
  if $|m_0| \neq |m_1|$
      return 0
  return $b \overset{?}{=} b'$
Proc. $\mathrm{KDerO}_{\mathrm{msk}}(x)$:
  $\mathrm{L} \leftarrow \mathrm{L} \cup \{x\}$
  $\mathrm{sk}_x \leftarrow_\$ \mathsf{IBE.KDer}(\mathrm{msk}, x)$
  return $\mathrm{sk}_x$
```

```
$a\mathsf{IND} - \mathsf{FE} - \mathsf{CPA}^{\mathcal{A}}_{\mathsf{FE}}(\lambda)$:
  $b \leftarrow_\$ \{0,1\}$
  $\mathrm{L} \leftarrow \emptyset$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow_\$ \mathsf{FE.Setup}(\lambda)$
  $(m_0, m_1) \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathrm{mpk})$
  $\mathsf{CT} \leftarrow_\$ \mathsf{FE.Enc}(\mathrm{mpk}, m_b)$
  $b' \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) \neq \mathcal{F}(k, m_1)$
      return 0
  return $b \overset{?}{=} b'$

Proc. $\mathrm{KDerO}_{\mathrm{msk}}(k)$:
  $\mathrm{L} \leftarrow \mathrm{L} \cup \{k\}$
  $\mathrm{sk}_k \leftarrow_\$ \mathsf{FE.KDer}(\mathrm{msk}, k)$
  return $\mathrm{sk}_k$
```

Figure 38.2: On the left, the indistinguishability chosen-plaintext experiment defined for an identity-based encryption scheme. On the right, the adaptive indistinguishability chosen-plaintext experiment defined for a functional encryption scheme.

The described indistinguishability notion for identity-based encryption limits the adversary to not being able to trivially decrypt the challenge messages. The peculiarity is that, in terms of functional encryption, only queries for keys evaluating to the error symbol are allowed. Indeed, only queries such that $x \neq x^*$ or, in other words, $\mathcal{F}(x, (x^*, m_0)) = \perp$ are valid. Therefore, the classic early abort condition for functional encryption stating that $\mathcal{F}(x_0, (x^*, m_0)) \neq \mathcal{F}(x_1, (x^*, m_0))$ is never satisfied. The negative effect of this restriction is that the weaker indistinguishability notion in Figure 38.1 is not compatible with identity-based encryption as the removal of error queries removes every query possibility from the adversary.

38.2.2 Attribute-based encryption

Attribute-based encryption generalizes identity-based encryption by not only allowing a single identity, but a set of attributes validating decryption. The formal definition below stems from [GKP+13].

Definition 38.3 (Attribute-Based Encryption)**.** An attribute-based encryption scheme (ABE) for a class of predicates $P \in \{\mathcal{P}_n\}_{n\in\mathbb{N}}$ and an associated message space $\mathcal{M}_0$ is a quadruple of algorithms ($\mathsf{ABE.Setup}$, $\mathsf{ABE.KDer}$, $\mathsf{ABE.Enc}$, $\mathsf{ABE.Dec}$) such that:

1. $(\mathrm{msk}, \mathrm{mpk}) \leftarrow_{\$} \mathsf{ABE.Setup}(\lambda)$: given a security parameter λ, it outputs a pair of master secret/public keys.

2. $\mathsf{CT} \leftarrow_{\$} \mathsf{ABE.Enc}(\mathrm{mpk}, x, m)$: the randomized encryption procedure encrypts the plaintext $m \in \mathcal{M}_0$ with respect to an attribute $x \in \{0,1\}^n$ $(n \in \mathbb{N})$ and the master public key mpk.

3. $\mathrm{sk}_P \leftarrow_{\$} \mathsf{ABE.KDer}(\mathrm{msk}, P)$: using the master secret key msk, the (possibly randomized) key-derivation procedure outputs a functional key sk_P corresponding to the predicate $P \in \{\mathcal{P}_n\}_{n\in\mathbb{N}}$.

4. $y \leftarrow \mathsf{ABE.Dec}(\mathrm{sk}_P, \mathsf{CT})$ decrypts the ciphertext CT using a predicate key sk_P to learn a message $m \in \mathcal{M}_0$ or the special error symbol $\perp$.

An ABE scheme is correct if for all messages $m \in \mathcal{M}_0$, all predicates $P \in \{\mathcal{P}_n\}_{n\in\mathbb{N}}$, and all attributes $x \in \{0,1\}^n$ such that $P(x) = 1$, the following holds:

$$\Pr\left[\, y = m \;\middle|\; \begin{array}{l} (\mathrm{msk}, \mathrm{mpk}) \leftarrow_{\$} \mathsf{ABE.Setup}(\lambda) \wedge \\ \mathsf{CT} \leftarrow_{\$} \mathsf{ABE.Enc}(\mathrm{mpk}, x, m) \wedge \\ \mathrm{sk}_P \leftarrow_{\$} \mathsf{ABE.KDer}(\mathrm{msk}, P) \wedge \\ y \leftarrow \mathsf{ABE.Dec}(\mathrm{sk}_P, \mathsf{CT}) \end{array} \right] \in 1 - \textsc{Negl}(\lambda)\ .$$

Remark 38.4. *Contrary to [GKP+13], we point out that correctness makes no requirement when $P(x) \neq 1$ [GVW15].*

Attribute-based encryption is a special case of predicate encryption where the predicate key space $\mathcal{K}_0$ is a set of Boolean formulas that evaluate the vector attributes $\Omega = R^n$. Thus, we may compare its traditional indistinguishability game [GKP+13] in Figure 38.3 with the original one for functional encryption.

The adversary is again limited to not decrypting the challenge messages. Furthermore, in line with the identity-based encryption game, we observe that only queries for keys evaluating to the error symbol are allowed. Indeed, only queries such that $P(x) = 0$ or, in other words, $\mathcal{F}(P, (x, m_0)) = \mathcal{F}(P, (x, m_1)) =\perp$ are valid. Thus, if $x^* \notin \{x_0, x_1\}$, then the classic early abort condition for functional encryption stating that $\mathcal{F}(x_0, (x^*, m_0)) \neq$

```
$\underline{\mathsf{IND-ABE-CPA}^{\mathcal{A}}_{\mathsf{ABE}}(\lambda)}$:
  $b \leftarrow_\$ \{0,1\}$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow_\$ \mathsf{ABE.Setup}(\lambda)$
  $P \leftarrow_\$ \mathcal{A}(\lambda, \mathrm{mpk})$
  $\mathrm{sk}_P \leftarrow_\$ \mathsf{ABE}.\mathrm{KDerO}_{\mathrm{msk}}(\mathrm{msk}, P)$
  $(m_0, m_1, x) \leftarrow_\$ \mathcal{A}(\mathrm{sk}_P)$
  $\mathsf{CT} \leftarrow_\$ \mathsf{ABE.Enc}(\mathrm{mpk}, x, m_b)$
  $b' \leftarrow_\$ \mathcal{A}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $|m_0| = |m_1|$ and $P(x) = 0$
      return $b \overset{?}{=} b'$
  return $0$
```

```
$\underline{a\mathsf{IND-FE-CPA}^{\mathcal{A}}_{\mathsf{FE}}(\lambda)}$:
  $b \leftarrow_\$ \{0,1\}$
  $\mathrm{L} \leftarrow \emptyset$
  $(\mathrm{mpk}, \mathrm{msk}) \leftarrow_\$ \mathsf{FE.Setup}(\lambda)$
  $(m_0, m_1) \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathrm{mpk})$
  $\mathsf{CT} \leftarrow_\$ \mathsf{FE.Enc}(\mathrm{mpk}, m_b)$
  $b' \leftarrow_\$ \mathcal{A}^{\mathrm{KDerO}_{\mathrm{msk}}(\cdot)}(\lambda, \mathsf{CT}, \mathrm{mpk})$
  if $\exists k \in \mathrm{L}$ s.t. $\mathcal{F}(k, m_0) \neq \mathcal{F}(k, m_1)$
      return $0$
  return $b \overset{?}{=} b'$

$\underline{\text{Proc. KDerO}_{\mathrm{msk}}(k)}$:
  $\mathrm{L} \leftarrow \mathrm{L} \cup \{k\}$
  $\mathrm{sk}_k \leftarrow_\$ \mathsf{FE.KDer}(\mathrm{msk}, k)$
  return $\mathrm{sk}_k$
```

Figure 38.3: On the left, the indistinguishability chosen-plaintext experiment defined for an attribute-based encryption scheme. On the right, the adaptive indistinguishability chosen-plaintext experiment defined for a functional encryption scheme.

$\mathcal{F}(x_1, (x^*, m_0))$ is never satisfied. Again, the negative effect of this restriction is that the weaker indistinguishability notion in Figure 38.1 is not compatible with attribute-based encryption, as the removal of error queries removes all query possibilities from the adversary.

38.2.3 Incompatibility with predicate encryption

Analysing the indistinguishability security game of identity-based encryption and attribute-based encryption, we note that none of them formally contradicts our attack, but they do not favour it either. As we did not manage to find vulnerable schemes, we believe that our attack is no threat to such primitives in practice. This accentuates the negative impact which would have the pragmatic change of Figure 38.1 as it compromises the well-established traditional security games by removing every query possibility from an adversary. Thereby, we conclude that this pragmatic patch is not universal enough to capture the right level of security for all functionalities of interest.

Chapter 39

Open Questions

Secure functional encryption schemes are difficult to achieve and the corresponding security notions deviate from their classical equivalents. Our development outlines a new threat and indicates how to avoid it. It is important to monitor flaws that emerge from new cryptographic primitives and to give best-practice examples for constructing secure schemes. Our new perspective on the error symbol and its implementation may be used for other security analyses leading to necessary criteria for secure functional encryption schemes.

Our study focussed on the impact of our attack on DDH-based functional encryption schemes for the bounded-norm inner-product functionality. However, its applicability has a wider range. A systematic literary review may filter out other vulnerable functional encryption schemes and may retrieve other affected functionalities. The same comment holds for other cryptographic primitives.

We believe that it is important to further investigate the special role of the error symbol in functional encryption. For example, in case of an error evaluation, correctness in functional encryption does not mandate the decryption procedure to output an error (see Section 35.1). Outputting a random value instead of an error may have an impact on the interpretation of the output, but also on the security notions. As our study in Chapter 38 shows, the error symbol is of the highest importance for some sub-primitives of functional encryption, and cannot be avoided in the definition, yet it needs to be handled with care.

Bibliography V

[AAB$^+$13] Shashank Agrawal, Shweta Agrawal, Saikrishna Badrinarayanan, Abishek Kumarasubramanian, Manoj Prabhakaran, and Amit Sahai. Functional encryption and property preserving encryption: New definitions and positive results. Cryptology ePrint Archive, Report 2013/744, 2013. https://ia.cr/2013/744.

[ABDCP15] Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In Jonathan Katz, editor, *Public-Key Cryptography – PKC 2015*, pages 733–751, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[ABN10] Michel Abdalla, Mihir Bellare, and Gregory Neven. Robust encryption. In Daniele Micciancio, editor, *Theory of Cryptography*, pages 480–497, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[ACF$^+$17] Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: Function-hiding realizations and constructions without pairings. Cryptology ePrint Archive, Report 2017/972, 2017. https://ia.cr/2017/972.

[AFH$^+$20] Martin Albrecht, Pooya Farshim, Shuai Han, Dennis Hofheinz, Enrique Larraia, and K.G. Paterson. Multilinear maps from obfuscation. *Journal of Cryptology*, 33, 01 2020.

[AJ15] Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In Rosario Gennaro and Matthew Robshaw, editors, *Advances in Cryp-*

tology – CRYPTO 2015, pages 308–326, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[ALS16] Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016*, pages 333–362, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[BBL17] Fabrice Benhamouda, Florian Bourse, and Helger Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In *PKC (2)*, pages 36–66. Springer, 2017.

[BCFG17] Carmen Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. pages 67–98, 07 2017.

[BGG$^+$14] Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit ABE and compact garbled circuits. In Phong Q. Nguyen and Elisabeth Oswald, editors, *Advances in Cryptology – EUROCRYPT 2014*, pages 533–556, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[BGI14] Elette Boyle, Shafi Goldwasser, and Ioana Ivan. Functional signatures and pseudorandom functions. In Hugo Krawczyk, editor, *Public-Key Cryptography – PKC 2014*, pages 501–519, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[BJK15] Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. volume 9452, pages 470–491, 11 2015.

[BKKW21] Christian Badertscher, Aggelos Kiayias, Markulf Kohlweiss, and Hendrik Waldner. Consistency for functional encryption. pages 1–16, 06 2021.

[BKS16] Zvika Brakerski, Ilan Komargodski, and Gil Segev. Multi-input functional encryption in the private-key setting: Stronger security from weaker assumptions. In *Proceedings, Part II, of the*

35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9666, page 852–880, Berlin, Heidelberg, 2016. Springer-Verlag.

[BO13] Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. pages 218–234, 11 2013.

[BRS13] Dan Boneh, Ananth Raghunathan, and Gil Segev. Function-private identity-based encryption: Hiding the function in functional encryption. *IACR Cryptol. ePrint Arch.*, 2013:283, 2013.

[BS15] Zvika Brakerski and Gil Segev. Function-private functional encryption in the private-key setting. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography*, pages 306–324, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *Theory of Cryptography*, pages 253–273, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[BV18] Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. *J. ACM*, 65(6), nov 2018.

[CLT18] Guilhem Castagnos, Fabien Laguillaumie, and Ida Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In Thomas Peyrin and Steven Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018*, pages 733–764, Cham, 2018. Springer International Publishing.

[DDM16] Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Functional encryption for inner product with full function privacy. In Chen-Mou Cheng, Kai-Min Chung, Giuseppe Persiano, and Bo-Yin Yang, editors, *Public-Key Cryptography – PKC 2016*, pages 164–195, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[DG21] Nico Döttling and Sanjam Garg. Identity-based encryption from the Diffie-Hellman assumption. *J. ACM*, 68(3), mar 2021.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DS00] Stefan Dierneder and Rudolf Scheidl. Conceptual design, functional decomposition, mathematical modelling, and perturbation analysis. In Peter Kopacek, Roberto Moreno-Díaz, and Franz Pichler, editors, *Computer Aided Systems Theory - EUROCAST'99*, pages 38–45, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[FLPQ13] Pooya Farshim, Benoît Libert, Kenneth G. Paterson, and Elizabeth A. Quaglia. Robust encryption, revisited. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *Public-Key Cryptography – PKC 2013*, pages 352–368, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY, USA, 2009. Association for Computing Machinery.

[GGH$^+$13] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 40–49, 2013.

[GGHZ16] Sanjam Garg, Craig Gentry, Shai Halevi, and Mark Zhandry. Functional encryption without obfuscation. In *TCC*, 2016.

[GKP$^+$13] Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 555–564, New York, NY, USA, 2013. Association for Computing Machinery.

[GVW15] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *J. ACM*, 62(6), dec 2015.

[JLS21] Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, page 60–73, 2021.

[KLZ96] Dexter Kozen, Susan Landau, and Richard Zippel. Decomposition of algebraic functions. *J. Symb. Comput.*, 22:235–246, 01 1996.

[KS17] Ilan Komargodski and Gil Segev. From minicrypt to obfustopia via private-key functional encryption. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 122–151, Cham, 2017. Springer International Publishing.

[KSW08] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, pages 146–162, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Lin16] Huijia Lin. Indistinguishability obfuscation from constant-degree graded encoding schemes. In *Proceedings, Part I, of the 35th Annual International Conference on Advances in Cryptology — EUROCRYPT 2016 - Volume 9665*, page 28–57, Berlin, Heidelberg, 2016. Springer-Verlag.

[LPST16] Huijia Lin, Rafael Pass, Karn Seth, and Sidharth Telang. Indistinguishability obfuscation with non-trivial efficiency. In *Proceedings, Part II, of the 19th IACR International Conference on Public-Key Cryptography — PKC 2016 - Volume 9615*, page 447–462, Berlin, Heidelberg, 2016. Springer-Verlag.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. https://ia.cr/2010/556.

[Sha85] Adi Shamir. Identity-based cryptosystems and signature schemes. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, pages 47–53, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[SW04] Amit Sahai and Brent Waters. Fuzzy identity based encryption. Cryptology ePrint Archive, Report 2004/086, 2004. https://ia.cr/2004/086.

[Tom19] Junichi Tomida. Tightly secure inner product functional encryption: Multi-input and function-hiding constructions. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology - ASIACRYPT 2019 - 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, December 8-12, 2019, Proceedings, Part III*, volume 11923 of *Lecture Notes in Computer Science*, pages 459–488. Springer, 2019.

Ending act: The future

Jay was staggered by the wisdom and the broad knowledge of his pops. He enjoyed every second of their conversation and listened carefully to the elderly advice. “Jay”, it echoed through the staircase, “it’s time for bed!” It was barely ten o’clock, but Jay’s mother knew that he would oversleep if he does not get his nine hours of sleep. “I’m coming”, shouted Jay. He hugged his pops and thanked him for the nice time.

He ran down the stairs and started his sleeping ritual: taking a shower, putting pyjamas on, washing his teeth, saying good night to everyone, sticking the tongue out to Mrs Skizzles, preparing his knapsack for school, and making himself comfortable in bed. Before sleep, he read a comic, as usual. The story was about his favourite character, Uncle Scrooge, and how he gained his first dime. Then, he turned off the lights and went to sleep.

Jay was lying there and reflected on his achievements: he mastered the numismatic essentials, he managed to help in the big count, albeit he needed to carry out the work twice, and he got to know about encryption procedures and their subtleties. However, Jay was not satisfied. He knew that he could do more, he knew that he could still improve. “One day”, he thought, “I will even outsmart pops and then I will be able to teach him something new.” It was not clear on which topic Jay would become a specialist nor when he would reach his goal, but Jay would not give up until he did. “In the future”, he figured out, “I still have a lot to learn. Until then, I will simply tackle one problem at a time, and see where it leads me to...”