Technische Universität München

TUM School of Computation, Information and Technology

Information-Theoretic Security Analysis and Coding with Feedback for Selected Channels

Georg Johannes Maringer

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitz:		Prof. Dr.-Ing. Eckehard Steinbach
Prüfende der Dissertation:	1.	Prof. Dr.-Ing. Antonia Wachter-Zeh
	2.	Assistant Prof. Dr. Alejandro Cohen

Die Dissertation wurde am 25.09.2024 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 24.02.2024 angenommen.

Abstract

This thesis consists of two parts. Part I deals with the idea of using information-theoretic concepts to improve the parameters of public key encryption schemes that base their security on the hardness of certain well studied lattice problems. We show that it is possible to interpret the concatenation of encryption and decryption of learning with errors based public key encryption schemes as data transmission over a noisy channel. We then use standard information-theoretic concepts to obtain a lower bound on the capacity of this channel and show how to change the public key encryption schemes such that either the amount of plaintext bits per ciphertext bit is increased or such that keysizes and ciphertext sizes are reduced for a constant plaintext size. Those modifications do not decrease the security level of the scheme. We also use finite blocklength information theory to show that the aforementioned goals can also be achieved under reasonable assumptions that we are stating in this work.

The other main topic in Part I of this thesis are Physical Unclonable Functions (PUFs) which are used for key derivation and tamper protection. The PUF foil we are investigating contains a capacitive mesh that can be used to protect an entire printed circuit board with controllers processing sensitive information from side-channel attacks. Assuming that an attacker has to drill a small hole into the foil that destroys a certain amount of capacitive cells. We investigate how many capacitive cells the foil needs to contain if a zero leakage helper data scheme shall be used in this setup. We obtain upper and lower bounds and show in that way that an existing setup either leaks information about the secret via the helper data or cannot achieve the claimed security parameters for other reasons.

Part II of this thesis is devoted to feedback coding. More specifically we consider noiseless instantaneous feedback, meaning that after each symbol transmission over the respective forward channel the encoder is notified about the received symbol at the decoder. Within a block transmission, the feedback gives the encoder the possibility to react according to the received symbols at the decoder. For probabilistic channels Shannon has shown that noiseless feedback does not increase the channel capacity. However, in this thesis we analyze the capacity error function (maximal achievable rate) in a combinatorial error model where the maximal number of errors scales linearly with the blocklength. We obtain upper and lower bounds on the capacity error function of the Z-channel, unidirectional channels and the binary insertion-deletion channel. For the binary insertion-deletion channel upper and lower bound are matching and therefore the capacity error function has been fully characterized.

Acknowledgments

First and foremost I would like to thank my supervisor Prof. Antonia Wachter-Zeh for her support and all the helpful discussions throughout my PhD at TUM. I would like to thank her for showing interest into my ideas and providing a different view on research problems that proved to be insightful many times. Furthermore, I would like to thank her for encouraging my collaborations with many outstanding researchers and in particular I would like to thank her for supporting my research visits in Gothenburg and Haifa.

I would like to thank Prof. Giuseppe Durisi for hosting me twice in Gothenburg and teaching me about various topics in Finite Blocklength Information Theory. Apart from his guidance at the universtiy I would like to thank him for showing me the Swedish lifestyle by going cross-country skiing with me.

Furthermore, I would like to thank Prof. Alejandro Cohen for hosting me at the Technion in Haifa. I would like to thank him for all the effort he made to help me finding my way at the Technion and of course for many helpful discussions on topics related at the intersection between cryptographic and information theoretic security. Moreover, I would like to thank him for agreeing to be part of the examination committee of this doctoral thesis.

I thank Prof. Josef A. Nossek for agreeing to become my mentor and for giving me valuable advice on several occasions throughout my PhD.

I would also like to thank Prof. Eckehard Steinbach for chairing my examination committee.

I spent a lot of time during the last years at the Institute for Communications Engineering (ICE) at TUM. I would like to thank all members of the institute for creating a nice working environment and all the discussions about scientific and non-scientific topics. Especially I would like to thank my office mates Haider Al Kim, Nikita Polyanskii, Vivian Papadopoulou and Hugo Sauerbier Couvée. I would like to thank Antonia, Prof. Gerhard Kramer and Prof. Norbert Hanik for creating such a nice working environment here at the institute, especially hosting many workshops and conferences here at the institute giving its members the opportunity to get in touch with many fantastic researchers coming from all over the world. Special thanks also goes to the non-scientific staff here at the institute. Without their work the institute could not have functioned nearly as well as it did throughout my PhD.

I would like to thank all my co-authors for their collaboration. I really enjoyed doing research with them: Christian Deppe, Gökberk Erdogan, Tim Fritzmann, Kathrin

Garb, Matthias Hiller, Thomas Jerkovits, Patrick Karl, Johannes Kunz, Ludwig Kürzinger, Vladimir Lebedev, Hedongliang Liu, Nikita Polyanskii, Sven Puchinger, Julian Renner, Stefan Ritterhoff, Thomas Schamberger, Johanna Sepúlveda, Georg Sigl, Ilya Vorobyev, Antonia Wachter-Zeh, Violetta Weger, Lorenz Welter, Marvin Xhemrishi.

I would also like to thank Maria Abu Sini, Matthias Hiller, Mahdi Mahvari, Nikita Polyanskii and Stefan Ritterhoff for proofreading parts of this thesis.

I thank Thomas Wiegart for many scientific and non-scientific discussions and most importantly for being such a good flatmate throughout my entire PhD time at TUM.

Last but not least, I would like to thank my mother Elfriede, my father Georg and my sister Christina for their love and support throughout my whole life. I would also like to thank my friends for their support and for reminding me of many important things in life apart from science.

Georg Johannes Maringer
Munich, August 2025

Contents

1 Introduction and Thesis Outline **1**

I Analysis of Communication Channels Related to Security

2 Preliminaries **7**
2.1 Notation 7
2.2 Background in Information Theory 9
2.3 Finite Blocklength Information Theory 19
2.4 Background in Coding Theory 23
2.5 Basics of Cryptography 27
2.6 Post-Quantum Cryptography 32
2.7 Physical Unclonable Functions 36

3 Information and Coding Theoretic Analysis of Learning With Errors Based Cryptosystems **43**
3.1 Introduction 44
3.2 RLWE/MLWE-Based Cryptography 46
3.3 The RLWE/MLWE Channel with Increased Alphabet Size 52
3.4 Information-Theoretic Analysis of the RLWE/MLWE Channel 57
3.5 Maximizing Asymptotic Rates for NewHope and Kyber 62
3.6 Minimizing Ciphertext and Key Sizes in the Asymptotic Setting 64
3.7 Achievable Data Rates and DFRs for a Single Ciphertext Block 66
3.8 Finite-Length Achievability Bounds 69
3.9 Summary and Open Problems 71

Appendices **73**
3.A Conditions of Corollary 3.2 for NewHope 73
3.B Conditions of Lemma 3.2 for Kyber 74
3.C Results for Frodo and LAC 75

4 Information Theoretic Analysis of PUF-Based Tamper Protection **81**
4.1 Introduction 81

4.2 State of the Art . . . 83
4.3 PUF-Based Tamper Protection Foil . . . 86
4.4 Secret Key Capacities for the Foil PUF . . . 90
4.5 Secret Sharing using One-Way Communication for Finite Lengths . . . 97
4.6 Achievability and Converse Bounds for Finite Lengths . . . 101
4.7 Applying Converse Results in Security Analysis . . . 108
4.8 Summary and Open Problems . . . 109

II Coding with Feedback for Selected Channels

5 Preliminaries 113
5.1 Notation . . . 113
5.2 Distance Metrics . . . 113
5.3 Coding with Feedback . . . 114

6 The Z-Channel 117
6.1 Introduction . . . 117
6.2 Upper Bound on $C_2^f(\Gamma_Z^2, \tau)$ . . . 119
6.3 Lower Bound on $C_2^f(\Gamma_Z^2, \tau)$ . . . 121
6.4 Tightness of Upper and Lower Bound and Comparison to Previous Results 133
6.5 Summary and Open Problems . . . 136

7 Unidirectional Channels 137
7.1 Introduction and Motivation . . . 137
7.2 Chapter Outline . . . 138
7.3 Symmetric, Asymmetric and Unidirectional Errors . . . 139
7.4 Zero-Error Capacity of Asymmetric and Unidirectional Channels . . . 140
7.5 An Upper Bound on Capacity Error Functions . . . 144
7.6 Lower Bound on the Capacity Error Function . . . 146
7.7 Constructing Lower Bounds Using Rubber Methods . . . 148
7.8 Summary and Open Problems . . . 153

8 Insertion-Deletion Channel 155
8.1 Introduction . . . 155
8.2 Upper Bound on $C_2^f(\Gamma_{id}^2, \tau)$ . . . 159
8.3 Lower Bound on $C_2^f(\Gamma_{id}^2, \tau)$ . . . 160
8.4 Summary and Open Problems . . . 165

Appendices 167
8.A Analysis of the Feedback Substitution Code . . . 167

9 Further Publications by the Author 191

9.1 FuLeeca 191
9.2 Signature Codes for the Multiple Access Channel 192

10 Conclusion **193**

List of Figures

2.1 Degraded wiretap channel (top), wiretap channel (bottom) 13
2.2 Secret Sharing using Common Randomness 16
2.3 Simplified schematic of a key generation scheme based on a PUF. . . 37
2.4 Key enrollment and key reconstruction for analog PUFs 40

3.1 RLWE/MLWE channel 53
3.2 RLWE/MLWE and marginalized RLWE/MLWE channel (dashed) . 57
3.3 Lower bounds on the capacities of the RLWE/MLWE channels according to NewHope and Kyber 63
3.4 Lower bounds on the maximal amount of plaintextbits per ciphertextbits for the RLWE/MLWE channels of NewHope and Kyber . . 63
3.5 Lower bounds on the capacities of the MLWE channel for Kyber with $l = 3$ and $k = 14$ achieving the security of the Kyber1024 parameter set 65
3.6 Lower bounds on the capacities of the MLWE channels for enhanced ciphertext compression for the Kyber512 and Kyber1024 parameter . 65
3.7 RCU bound and normal approximation for the Kyber512 parameter set with $d_u = 7, d_v = 4$ and for the Kyber1024 parameter set with $d_u = 8, d_v = 3$, required DFRs for Kyber512 (black, dashed) and Kyber1024 (black) 71
3.8 RCU bound and normal approximation for Kyber with $l = 3$ and $k = 14$ achieving the security level of the Kyber1024 parameter set . 72
3.C.1 Lower bound on the capacity of the LWE channel according to Frodo 76
3.C.2 Lower bound on the capacity of the RLWE channel according to LAC (Round 1 submission) 77
3.C.3 Lower bounds on the maximal amount of plaintextbits per ciphertextbits for the LWE channels of Frodo 77
3.C.4 Lower bounds on the maximal amount of plaintextbits per ciphertextbits for the LWE channels of LAC (Round 1 submission) 78

4.2.1 Sketch of PUF-based envelope [IOK+18] with two conducting electrode layers in red and blue, and two shielding layers in black (figure copied with permission of Matthias Hiller) 85
4.3.1 Enrollment phase 89

4.3.2 Reconstruction phase 90
4.4.1 Key enrollment and key reconstruction for legitimate user, digital and analog attacker 91

5.3.1 Channel with feedback 114
5.3.2 Symmetric channel Γ_S^2 116

6.1.1 Z-channel Γ_Z^2 118
6.3.1 Encoding algorithm for transmission over the Z-channel 123
6.3.2 Partitioning algorithm 130
6.4.1 Bounds on the capacity error function for the Z-channel with noiseless feedback. 134

7.3.1 Binary Z-channel Γ_Z^2 and inverse Z-channel $\Gamma_{\overline{Z}}^2$ 139
7.3.2 Generalized Z-channel and generalized inverse Z-channel 141
7.5.1 Bipartite graph Γ^{q*} 144
7.6.1 Lower and upper bound on the capacity error function of the unidirectional channels Γ_U^5 and Γ_U^6 149
7.7.1 Comparison of the lower bound on $C_2^f(\Gamma_U^2, \tau)$ obtained by using the modified rubber method and the capacity error function of the symmetric channel $C_2^f(\Gamma_S^2, \tau)$ 154

8.1.1 Capacity error function of the insertion-deletion channel with noiseless feedback. 158
8.A.1 Illustration for the notions of a crossing, a balanced and central true segment and the corresponding areas when $\tau = 0.15, \alpha = 2\tau$, and $\beta = 2 - \alpha$. 169
8.A.2 Example for transition of the intervals $L(i), T(i)$, and $R(i)$ in case for a correct and incorrect transmission 169
8.A.3 Illustration for level sets of the function $v(i)$ for $\tau = 0.15$, $\alpha = 2\tau$, and $\beta = 2 - \alpha$. 172
8.A.4 Illustration for level sets of the function $u(i)$ for $\tau = 0.15$, $\alpha = 2\tau$, and $\beta = 2 - \alpha$. 172
8.A.5 Illustration of the proof strategy for Theorem 8.2. For the inequality (8.4), Proposition 8.5, or Proposition 8.6 the arrows are omitted due to the sake of clarity. The small indicator boxes however show where they are used. 174
8.A.6 Illustration for a trajectory of $t(i) = 1 - l(i) - r(i)$. We have $M = 2^{14}$ and transmit a random message, where we use $n' = 27$ total transmissions. Errors occur at time steps 16, 21, and 22. Used parameters are $\tau = 0.15$, $\alpha = 2\tau$, and $\beta = 2 - \alpha$. 175

List of Tables

3.1 Code parameters for different Q for NewHope with DFR $< 2^{-216}$, $(n, q, k) = (1024, 12289, 8)$ 67

3.2 Code parameters for different Q for Kyber with DFR $< 2^{-174}$, $(n, q, k, l) = (256, 3329, 2, 4)$ 67

3.3 Decryption failure rates for different alphabet sizes Q for a BCH-bitrate of at least 0.25 for NewHope and 1 for Kyber 68

3.4 Code parameters for different Q for Kyber with DFR $< 2^{-174}$, $(n, q, s, l) = (256, 3329, 2, 4)$ for coding over 4 blocks 69

3.5 Code parameters for different Q for NewHope with DFR $< 2^{-216}$, $(n, q, k) = (1024, 12289, 8)$ for coding over 4 blocks 70

3.C.1 Code parameters for different Q for Frodo with DFR $< 2^{-252.5}$, $(n, q, s, l) = (1, 65536, 6, 8)$ 78

3.C.2 Code parameters for different Q for LAC256 and DFR $< 2^{-115.4}$ . . . 78

3.C.3 Code parameters for different Q for Frodo with DFR $< 2^{-252.5}$, $(n, q, s, l) = (1, 65536, 6, 8)$ for coding over 4 blocks 79

3.C.4 Code parameters for different Q for LAC256 and DFR $< 2^{-115.4}$ for coding over 4 blocks 79

4.4.1 Achievable asymptotic rates for the PUF-channel for weak and strong attackers, different input quantization alphabets and quantization strategies, erasure probability $p_d = 0.1$ for the digital attacker and $p_a = 0.2$ for the analog attacker, $\sigma_P = 2241, \sigma_N = 129$ 95

4.4.2 Achievable asymptotic rates for the PUF-channel for weak and strong attackers, different input quantization alphabets and quantization strategies, erasure probability $p_d = 0.18$ for the digital attacker and $p_a = 0.36$ for the analog attacker, $\sigma_P = 2241, \sigma_N = 129$ 96

4.6.1 Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p_d = 0.1$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization, $\sigma_P = 2241, \sigma_N = 129$ 104

4.6.2 Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p_d = 0.18$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization, $\sigma_P = 2241, \sigma_N = 129$ 104
4.6.3 Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p = 0.1$, PUF reliability $\varepsilon = 10^{-9}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization 105
4.6.4 Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p = 0.18$, PUF reliability $\varepsilon = 10^{-9}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization 105
4.6.5 Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p_d = 0.18$ and $p_a = 0.36$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, analog attacker, equiprobable quantization 108
4.6.6 Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p = 0.18$ and $p_a = 0.36$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, analog attacker, equidistant quantization 108

Abbreviations

BCH	Bose-Chaudhuri-Hocquenghem
BEC	Binary Erasure Channel
BSC	Binary Symmetric Channel
CVP	Closest Vector Problem
DWTC	Degraded Wiretap Channel
EC	Erasure Channel
ECC	Error Correcting Code
GRS	Generalized Reed–Solomon
HDA	Helper Data Algorithm
iid	independent identically distributed
IoT	Internet of Things
LDPC	Low Density Parity Check
LWE	Learning With Errors
ML	Maximum Likelihood
MLWE	Module Learning With Errors
pdf	probability density function
pmf	probability mass function
PPT	Probabilistic Polynomial Time
PUF	Physical Unclonable Function
RCU	Random Coding Union
RLWE	Ring Learning With Errors
RV	Random Variable
RS	Reed–Solomon
SIVP	Shortest Independent Vector Problem
SRAM	Static Random Access Memory
SVP	Shortest Vector Problem
WTC	Wiretap Channel

1
Introduction and Thesis Outline

The interconnection of electronic devices has a huge impact on the daily life of human beings all over the world. Information can be gathered virtually everywhere using small IoT devices. Nowadays the access to information is arguably equally important as the access to natural resources like oil or metals. Moreover, networks exchanging data all over the world are used to steer our infrastructure and industries. Services like online payment are standard for the vast majority of people in our society. With the increasing relevance of the communication infrastructure it becomes important to protect services from malicious intruders. Attack scenarios are for instance eavesdropping of a network link to gain unauthorized access to private data. Furthermore, undetected modification of messages or impersonating can cause unintended actions by otherwise legitimately acting parties connected to the network. To prevent such attacks, protocols and cryptographic algorithms have been developed. Rather than trying to prevent an attacker from eavesdropping the communication, which is almost impossible for instance in wireless communication, the data transfer between the legitimate entities is encrypted. Digital signatures on the other hand can be used to ensure data integrity and authentication, thereby providing solutions to the other two security concerns.

The legitimate communication parties using a cryptographic scheme for encryption or authentication require some extra knowledge that a potential attacker does not have. The approach of keeping the design of the cryptographic scheme secret has failed the test of time. Therefore, the state of the art approach is to assume the legitimate parties to have an extra piece of information which commonly referred to as their private keys. Those keys need to be generated and stored in a way such that attackers cannot obtain access to them.

While it may be a reasonable assumption to assume that attackers cannot get physical access to a personal computer standing inside a secured building, for IoT devices and embedded systems this assumption is definitely not justified.

Secret keys need to be stored on the device and if an attacker can just read out the memory and accesses the private key, the use of cryptographic algorithms becomes

obsolete. Therefore, especially for embedded systems and IoT devices storing keys in a secure way is a major challenge for the designers of those devices. Having physical access to a chip enables an attacker to perform hardware measurements that leak information about secret data processed within cryptographic algorithms. Prominent examples of the measured quantities that may leak secret information to an attacker are runtime [Koc96; DKL+00; Sch00], power consumption [KJJ99; MOP08] or electromagnetic radiation [RR01]. It is even possible to change the operating conditions of the device such that the usual program flow is altered or errors are injected into computations [BDL97; BBKN12; BH22], e.g., by laser fault injection. Those kinds of attacks are subsumed under the term side-channel attacks.

Part I of this thesis is devoted to the interpretation of security related topics in the form of noisy communication channels. Once the channels of relevance have been specified, it is possible to use information- and coding-theoretic tools to analyze properties like channel capacity and their block error probabilities. The channels investigated in this work naturally arise from lattice-based public-key cryptography [Reg05; LPR10; BGV14] as well as physical unclonable functions (PUFs) [IOK+18; GXKF22a].

In Chapter 2 we introduce notation used in Part I of the thesis, information- and coding theoretic background, fundamental cryptographic concepts and the Learning with Errors problem including some variants of it. Furthermore, we provide an introduction on PUFs and develop several basic results on helper data algorithms used in the PUF context.

In Chapter 3 we first motivate why the concatenation of encryption and decryption within Learning With Errors (LWE) based public key encryption schemes can be interpreted as data transmission over a noisy communication channel. Those public key encryption schemes have received strong interest by the cryptographic community because they are expected to be secure even if an attacker has access to a large scale quantum computer. We develop a lower bound on the achievable rate of the resulting channel in the asymptotic as well as the finite blocklength setting. Furthermore, we present BCH codes [BRC60] to obtain practically achievable results under the assumption of stochastically independent coefficient failures within the resulting channel. We are interested in two different but related tasks. The first one is to maximize the number of plaintext bits per ciphertext bit to maximize the throughput. This is of relevance if the LWE based algorithm is used to exchange large amounts of secret data between the legitimate users on its own. The second one is the minimization of key and ciphertext sizes for a fixed amount of plaintext. The relevance of this attempt stems from the fact that public key algorithms usually enable the two parties to share a key of fixed size, e.g., 256 bit, that is then used within a symmetric encryption scheme to secure the communication for the payload.

The other security-related topic discussed in this work are Physical Unclonable Functions [PRTG02; HYKD14; Mae12] which are examined in Chapter 4. Their applications are twofold. They can be used to uniquely identify a device, similar to

fingerprint detection or retina scans for human beings, and as secure storages of private keys on devices. To establish this functionality the designer of the PUF uses minuscule manufactoring variations of the device. Furthermore, the Fraunhofer Institute for Applied and Integrated Security (AISEC) has developed a foil [IOK+18] consisting of capacitances that can be wrapped around circuity requiring protection from several sorts of side-channel attacks, e.g. laser or EM fault injection and EM or optical passive side-channels.

Part II of this thesis is devoted to coding with feedback. In this part of the thesis we are not considering probabilistic channels which is arguably the more commonly considered scenario in communication theory that has also been investigated by Shannon in his seminal work [Sha48]. However, notice that transmitting information at a rate below channel capacity does not guarantee correct decoding of the message at the receiver, it is only possible for the block error probability to be arbitrarily close to zero but not equal to zero. As an example, consider the binary symmetric channel with crossover probability $0 < p < 1/2$. Irrespective of its input, it is possible that the channel outputs the all ones vector and hence no guarantee for distinguishing even two messages at the decoder can be given.

Hence, Shannon introduced the notion of the zero-error capacity [Sha56] in contrast to nowadays more commonly used notion of channel capacity. This zero error capacity provides the maximal achievable rate for which an encoder-decoder pair exists such that any message m in the set of possible input messages $\mathcal{M}$ is guaranteed to be received correctly. The channel is in this case not specified by transition probabilities but rather in the more coarse way of specifying for each input symbol to which output symbols the channel is able to map it. The capacity error function is a generalisation of the zero error capacity for which the maximal amount of errors within a block of length n is specified. Throughout this work, we denote the maximal amount of errors within a block by the letter t. In this work, we investigate the asymptotic setting as $n \to \infty$ and we assume the maximal amount of errors to scale linearly with n, i.e. the maximal fraction of errors is defined as $\tau := t/n$. In particular the zero error capacity is equal to the capacity error function with parameter $\tau = 1$.

The capacity error function with noiseless instantaneous feedback for the symmetric channel with binary input has been determined for arbitrary $\tau \in [0, 1]$ by Berlekamp [Ber64] and Zigangirov [Zig76]. However, for many channels the capacity error function with feedback is still unknown. In this work, we consider particular classes of asymmetric channels, unidirectional channels and insertion deletion channels.

Chapter 5 introduces notation used in Part II of this thesis and basic definitions and concepts within the field of coding with feedback that are used throughout the subsequent chapters.

In Chapter 6 we present results for the Z-channel. This channel is asymmetric, meaning that not every input symbol can be mapped to every output symbol of the channel. Specifically, the zero symbol cannot be changed by the channel while a one at

the channel's input can be changed to a zero at the channel output. We provide lower and upper bounds on the capacity error function of the Z-channel. We conjecture that the lower bound is indeed tight.

In Chapter 7 we investigate the capacity error function for unidirectional channels with noiseless feedback. Unidirectional channels are composed of two special sorts of asymmetric channels. During each block transmission one of the two channels is selected without notifying the selection to transmitter or receiver und remains unchanged throughout the block. If an error occurs the sender knows which of the two channels has been selected and can adjust the encoding strategy accordingly. We provide upper and lower bounds on the capacity error function. Our results depend on the parity of the alphabet size of the channels. One encoding algorithm that we are using cannot be applied to channels with binary input and output alphabets. However, we were able to adapt the rubber method [Leb16] to unidirectional channels, thereby creating an encoding strategy also for binary alphabets.

In Chapter 8 we determine the capacity error function for the insertion-deletion channel with binary input over the full range of τ, i.e. $0 \leq \tau \leq 1$. In the proofs we make use of the known capacity error function for the symmetric channel with binary input and output alphabet. In Appendix 8.A we give a more elaborate proof for the capacity error of the symmetric channel, originally proposed by Zigangirov in [Zig76].

Chapter 9 provides a short summary on additional works the author has contributed to throughout his time as a doctoral researcher at the Institute for Communications Engineering. In particular short summaries on the digital signature scheme FuLeeca and a work on signature codes for the multiple adder channel are given.

Chapter 10 concludes the thesis by summing up results and giving an outlook on open problems related to the research presented throughout this thesis.

Part I

Analysis of Communication Channels Related to Security

2
Preliminaries

This chapter introduces the notation used throughout the first part of this work and reviews several well known concepts of cryptography, information theory and coding theory.

2.1 Notation

We denote scalars by lowercase letters and vectors by lowercase bold letters. Matrices are denoted by uppercase bold letters. We denote the i-th column of the matrix $\mathbf{A}$ by $\mathbf{a}_i$. Analogously, we denote matrices containing polynomials as their components by bold capital letters and denote vectors containing polynomials as their elements by bold lowercase letters and polynomials by lowercase letters, e.g., the matrix $\mathbf{A}$ with components from the polynomial ring $\mathcal{R}_q$ (formally defined later in Definition 2.2) with its i-th column $\mathbf{a}_i$ and the component in the i-th row and the j-th column by $a_{ij}(x) \in \mathcal{R}_q$ and frequently omit the indeterminate x if it is clear from context.

We denote random variables (RVs) by uppercase letters and their realizations by lowercase letters, i.e. the realization of a RV X is denoted by x. Furthermore, we denote the probability mass function of a discrete RV X by $P_X(x)$, the probability density function of a continuous RV X by $f_X(x)$ and the cumulative distribution by $F_X(x)$ in both cases. We denote the expectation operator by $\mathbb{E}[\cdot]$. If three random variables X, Y, Z form a Markov chain we write $X \multimap Y \multimap Z$.

Sampling an element b from a distribution χ is denoted by $b \xleftarrow{\$} \chi$ and sampling uniformly from a set $\mathcal{S}$ is denoted by $b \xleftarrow{\$} \mathcal{S}$. Independent sampling of every coefficient of a polynomial $a \in \mathcal{R}_q$ according to a distribution χ is denoted by $a \xleftarrow{\$} \chi(\mathcal{R}_q)$ and independent sampling of a vector $\boldsymbol{v} \in \mathcal{R}_q^l$ according to χ is denoted by $\boldsymbol{v} \xleftarrow{\$} \chi(\mathcal{R}_q^l)$. We denote the binomial distribution by $\mathcal{B}(i, n, p)$, where i specifies the number of successes, n the number of trials and p the success probability. We define the magnitude of an element in $\mathbb{Z}_q$ by the magnitude of its representation in the interval $[-q/2, q/2]$. We

denote the Gaussian distribution with mean μ and variance σ^2 by $\mathcal{N}(\mu, \sigma^2)$ and we write that a RV X is distributed according to a Gaussian by $X \sim \mathcal{N}(\mu, \sigma^2)$.

We denote the Gaussian Q-function by

$$Q(x) := \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z$$

and the complementary error function by

$$\operatorname{erfc}(x) := \frac{2}{\sqrt{\pi}} \int_x^\infty \exp(-z^2)\, \mathrm{d}z \ .$$

We define for a functions $f(n), g(n)$ the Landau symbols

$$f(n) \in o(g(n)) \Leftrightarrow \lim_{n\to\infty} \left|\tfrac{f(n)}{g(n)}\right| = 0$$

and

$$f(n) \in \mathcal{O}(g(n)) \Leftrightarrow \limsup_{n\to\infty} \left|\tfrac{f(n)}{g(n)}\right| < \infty \ .$$

The rounding operator is denoted by $\lceil . \rfloor$, where in particular $\lceil x.5 \rfloor = x + 1$.

Let $P_X * P_Y$ denote the convolution of two probability mass functions P_X and P_Y and let the n-fold convolution of P_X with itself be $\circledast_n(P_X) := P_X * P_X * \cdots * P_X$, in particular $\circledast_1(P_X) = P_X * P_X$ and $\circledast_0(P_X) = P_X$.

We denote sets by caligraphic letters, e.g., a set $\mathcal{S}$ and its cardinality by $|\mathcal{S}|$.

Definition 2.1 (Centered binomial distribution)**.** *The centered binomial distribution with parameter k, denoted as χ_k, is defined as $\chi_k(x) := \mathcal{B}(x + k/2, k, 1/2)$, where $x \in \{-k/2, -k/2 + 1, \ldots, k/2\}$.*

The expectation of the centered binomial distribution χ_k is 0 and its variance is $k/2$. It is possible to sample relatively efficiently from this distribution compared to, e.g., the rounded Gaussian distribution.

Definition 2.2. *Let $\mathcal{R}_q := \mathbb{Z}_q[x]/(x^n + 1)$ be the polynomial ring in x of degree n with coefficients in $\mathbb{Z}_q$. The addition of two polynomials in $\mathcal{R}_q$ is performed by adding the coefficients in $\mathbb{Z}_q$:*

$$a(x) + b(x) = c(x), \quad \textit{where } c_k = a_k + b_k \mod q.$$

The multiplication of two polynomials in $\mathcal{R}_q$ is defined by

$$a(x) \star b(x) = c(x), \ \textit{where } c_k = \sum_{i=0}^{k} a_i b_{k-i} - \sum_{i=k+1}^{n-1} a_i b_{n-i+k},$$

for all $k = 0, \ldots, n-1$. Thus, each polynomial in $\mathbb{Z}_q[x]/(x^n+1)$ can be represented by a polynomial in $\mathbb{Z}_q[x]$ of degree $< n$. Frequently throughout this work, we use ab as a shorthand notation for the multiplication of two polynomials $a, b \in \mathcal{R}_q$.

2.2 Background in Information Theory

Information theory is a scientific discipline that deals with mathematically representing information and has a wide range of applications. Two standard tasks are the representation of information in its most compact form, i.e. with as few symbols as possible, and the reliable transmission of data over potentially noisy communication channels. In his seminal work [Sha48] Shannon added redundancy within the encoder to enable the decoder to correct symbols that have been modified by the channel noise. The goal of a communication system designer is to construct encoders and decoders such that the block error probability and the required redundancy are minimized and the cardinality of the set of possible messages is maximized for a given blocklength n. These goals are fundamentally conflicting each other, e.g., one has to be prepared to shrink the cardinality of possible messages in order to achieve smaller block error probability. We start the section by defining several notions used throughout the section. Then we present a set of well-known theorems addressing the previously discussed basic problems of compression and reliable data transmission. Those have already been derived in [Sha48]. The results can for instance also be found in textbooks on information theory, e.g., [CT99].

Definition 2.3 (Encoder)**.** *We define an encoder $\mathcal{E} : \mathcal{M} \to \mathcal{X}^n$ to be a function that maps a message m_k from the set of possible messages $\mathcal{M}$ to the set of input sequences to a channel $P_{Y^n|X^n}$.*

Definition 2.4 (Decoder)**.** *We define a decoder $\mathcal{D} : \mathcal{Y}^* \to \mathcal{M} \cup \perp$ to be a function that maps a sequence of received symbols to the set of possible messages and an error symbol $\perp$.*

Definition 2.5 (Communication Channel)**.** *We define a communication channel (for short a channel) by the transition probabilities from the set of input sequences $\mathcal{X}^n$ to the set of output sequences $\mathcal{Y}^n$, i.e. the channel is specified by $P_{Y^n|X^n}(y^n|x^n)$ with $x^n \in \mathcal{X}^n$ and $y^n \in \mathcal{Y}^n$.*

Definition 2.6 (Memoryless channel)**.** *We say that a channel is memoryless if it holds that*

$$P_{Y^n|X^n}(\mathbf{y}^n|\mathbf{x}^n) = \prod_{i=1}^{n} P_{Y_i|X_i}(y_i|x_i) \ . \tag{2.1}$$

Definition 2.7. *We say that a channel created an error at the decoder's input if an input symbol x has been mapped to an output $y \neq x \in \mathcal{X}$. We define a symbol erasure*

to be the event that an input symbol x has been erased and hence this symbol gives the receiver no information about the channel input. Notice that the synchronization between channel input and channel output is not lost in this case as it is when symbols are deleted. If an erasure occurs at the i-th symbol we denote this by setting the random variable $Y_i = E$, where the event E denotes that an erasure occurred.

Definition 2.8. *We denote the average block error probability by*

$$\overline{P}_e := \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} P_e(m) \ , \tag{2.2}$$

where

$$P_e(m) = P_{Y^*|\mathcal{M}}(\mathcal{D}(Y^*) \neq m | m) \tag{2.3}$$

denotes the probability that an error occurs if the message m is transmitted. We denote the maximal error probability by

$$P_{e,max} = \max_{m \in \mathcal{M}} P_e(m) \ . \tag{2.4}$$

Definition 2.9. *We define a code $\mathcal{C}$ to be a subset of the set of possible channel input sequences $\mathcal{X}^n$. The parameter n is frequently referred to as the length of a code or simply the blocklength. We denote the cardinality of the code by $M := |\mathcal{C}|$. For a distance metric $d_{\mathcal{C}} : \mathcal{X}^n \times \mathcal{X}^n \to \mathbb{N}_0$, we define the minimum distance d of a code by*

$$d := \max_{c_1, c_2 \in \mathcal{C}} d_{\mathcal{C}}(c_1, c_2) \ . \tag{2.5}$$

For short we specify a code with blocklength n, having cardinality M and minimum distance d as an (M, n, d) code.

Definition 2.10. *We define the **rate** R of an (M, n, d) code to be*

$$R := \frac{1}{n} \log_2(M) \ . \tag{2.6}$$

Definition 2.11. *We define the support of a probability mass function P_X for a random variable X taking values in a set $\mathcal{X}$ by*

$$\operatorname{supp}(P_X) := \{x \in \mathcal{X} : P_X(x) > 0\} \ . \tag{2.7}$$

Definition 2.12. *We define the **entropy** of a discrete RV X to be*

$$H(X) := - \sum_{a \in \operatorname{supp}(P_X)} P_X(a) \log_2(P_X(a)) \tag{2.8}$$

and the conditional entropy of X given that some random variable $Z = z$ to be

$$H(X|Z=z) \coloneqq -\sum_{a \in \mathrm{supp}(P_{X|Z=z})} P_{X|Z}(a|z) \log_2(P_{X|Z}(a|z)) \ . \tag{2.9}$$

Furthermore, we denote the conditional entropy of X given Z to be

$$H(X|Z) \coloneqq -\sum_{(a,b) \in \mathrm{supp}(P_{XZ})} P_{XZ}(a,b) \log_2(P_{X|Z}(a|b)) = \sum_{b \in \mathrm{supp}(Z)} P_Z(b) H(X|Z=b) \ . \tag{2.10}$$

Definition 2.13. *We define the **differential entropy** of a continuous RV X with probability density function (pdf) f_X by*

$$h(X) \coloneqq -\int_{a \in \mathrm{supp}\, f_X} f_X(a) \log_2(f_X(a)) \, \mathrm{d}a \ . \tag{2.11}$$

The definition of the conditional differential entropy of X given some random variable Z is analogous to eq. (2.10) in Definition 2.12.

Definition 2.14. *We define the **variational distance** $d(P,Q)$ between two probability mass functions (pmfs) P and Q defined over the same domain $\mathcal{X}$ by*

$$d(P,Q) = \frac{1}{2} \sum_{x \in \mathcal{X}} |P(x) - Q(x)| \ . \tag{2.12}$$

Definition 2.15. *We specify the Kullback-Leibler divergence between two pmfs P and Q over the same domain by*

$$D(P||Q) = \sum_{x \in \mathcal{X}} P(x) \log_2 \left(\frac{P(x)}{Q(x)} \right) \ , \tag{2.13}$$

where we define $0 \log_2(0) := 0$.

Definition 2.16. *The mutual information between two random variables X and Y is specified by*

$$\begin{aligned} I(X;Y) &\coloneqq D(P_{XY}||P_X P_Y) \\ &= \sum_{(a,b) \in \mathrm{supp}(P_{XY})} P_{XY}(a,b) \log_2 \left(\frac{P_{XY}(a,b)}{P_X(a) P_Y(b)} \right) = H(Y) - H(Y|X) \ . \end{aligned} \tag{2.14}$$

If the random variable Y is continuous it holds that

$$I(X;Y) \coloneqq h(Y) - h(Y|X) \ . \tag{2.15}$$

Definition 2.17. *The mutual information $I(X;Y|Z=c)$ between two random variables X and Y conditioned on a discrete random variable Z taking the value c is defined by*

$$I(X;Y|Z=c) := H(X|Z=c) - H(X|Y,Z=c) \tag{2.16}$$

if X is discrete and

$$I(X;Y|Z=c) := h(X|Z=c) - h(X|Y,Z=c) \tag{2.17}$$

if X is continuous. Similar to Definition 2.12 we define

$$I(X;Y|Z) := \sum_{c \in \mathrm{supp}(P_Z)} P_Z(c)\, I(X;Y|Z=c) \ . \tag{2.18}$$

For continuous Z with pdf f_Z we define

$$I(X;Y|Z) = \int_{c \in \mathrm{supp}(f_Z)} f_Z(c)\, I(X;Y|Z=c)\, \mathrm{d}c \ . \tag{2.19}$$

Theorem 2.1 (Source coding theorem)**.** *Let a random variable X have entropy $H(X)$. Then a sequence of n independently identically distributed (i.i.d.) symbols according to the distribution of X can be losslessly compressed arbitrarily close to a sequence of $nH(X)$ bits with negligible risk of reconstruction failure as n goes to infinity.*

Theorem 2.2 (Channel coding theorem)**.** *Let a synchronized channel between a sender and a receiver be used for a block transmission of length n be specified by $P_{Y^n|X^n}$ over input alphabet $\mathcal{X}^n$ and output alphabet $\mathcal{Y}^n$. Then there exists a code $\mathcal{C}$ and an encoder/decoder pair with code rate arbitrarily close to C such that the block error probability of the communication system goes to 0 as $n \to \infty$, where C is specified to be*

$$C = \max_{P_{X^n}} \frac{1}{n} I(X^n;Y^n) \ . \tag{2.20}$$

For a memoryless channel it holds that

$$C = \max_{P_X} I(X;Y) \ . \tag{2.21}$$

Furthermore, the block error probability goes to 1 as $n \to \infty$ for any coding scheme with a rate greater than C.

Remark 2.1. *In case the random variable X is continuous the same statement holds if we maximize over the pdf f_{X^n} and f_X for equations (2.20) and (2.21), respectively.*

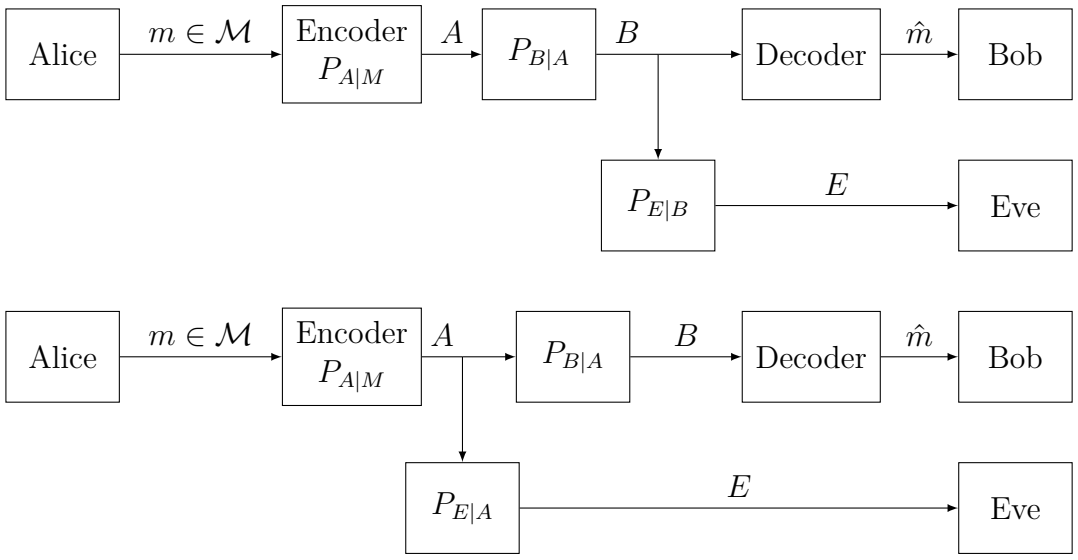

Figure 2.1: Degraded wiretap channel (top), wiretap channel (bottom)

2.2.1 The Wiretap Channel

In [Wyn75] Wyner introduced a channel model, in the following referred to as the Degraded Wiretap Channel (DWTC) (see also [BB11]). The channel has one input, in the following denoted by the random variable A, and two outputs B and E accessed by Bob and Eve, respectively. This is illustrated in the upper part of Fig. 2.1. The goal of Alice is to use this channel to reliably transmit a message m to Bob, i.e. $m = \hat{m}$ with high probability while keeping any information about the message secret from an eavesdropper called Eve. For the degraded case it is obvious that the channel from Alice to Bob has higher channel capacity compared to the one from Alice to Eve. The joint distribution of the channel input and its output is given as $P_{ABE}(a, b, e) = P_A(a)P_{B|A}(b|a)P_{E|B}(e|b)$. The same does not necessarily hold for the more general wiretap channel (WTC) studied by Csiszar and Körner in [CK78] and shown in the lower part in Fig. 2.1. Here, the eavesdropper's channel output E is not a degraded version of B rather it is directly generated from A through a noisy channel, i.e. $P_{ABE}(a, b, e) = P_A(a)P_{B|A}(b|a)P_{E|A}(e|a)$ in this case.

The next definition is a slightly adapted version of the definition of secrecy codes in [YSP19].

Definition 2.18. *An $(|\mathcal{M}|, \varepsilon, \delta)$ secrecy coding strategy for a (degraded) wiretap channel $(\mathcal{A}, P_{B,E|A}, \mathcal{B} \times \mathcal{E})$ consists of*

- *a set of possible messages $\mathcal{M} := \{1, \ldots, |\mathcal{M}|\}$ from which a message $M = m$ is selected,*

- *a randomized encoder that generates a codeword $A(m)$ for $m \in \mathcal{M}$ according to a pdf $P_{A|M=m}$ and*
- *a decoder $Dec : \mathcal{B} \to \mathcal{M}$ that assigns an estimate $\hat{M}$ to each received signal $B \in \mathcal{B}$.*

Encoder and decoder satisfy the average error probability (averaging performed by uniformly sampling the input message and over the randomness induced by the stochastic encoder)

$$Pr(Dec(B) \neq M) \leq \varepsilon$$

where for B it holds that $P_{B|M}(b|m) = \sum_{a \in \mathcal{A}} P_{B|A}(b|a) P_{A|M}(a|m)$. We distinguish average secrecy and maximum secrecy in the following way. For average secrecy with security parameter δ we require that

$$d(P_{ME}, P_M^{unif} P_E) \leq \delta \ . \tag{2.22}$$

where P_M^{unif} denotes the uniform distribution over the space of possible messages. In contrast for maximum secrecy it has to hold that

$$\max_{m \in \mathcal{M}} d(P_{E|M=m}, Q_E) \leq \delta \ , \tag{2.23}$$

where Q_E is the marginal distribution of E if uniformly distributed messages are transmitted over the channel. If we make the number of channel uses specific, we call an $(|\mathcal{M}|, \varepsilon, \delta)_{avg}$ average secrecy code for the channel $P_{B^n E^n|A^n}$ an $(n, |\mathcal{M}|, \varepsilon, \delta)_{avg}$ secrecy code. Codes for maximum secrecy are denoted by $(n, |\mathcal{M}|, \varepsilon, \delta)_{max}$. In the following we frequently omit specifying whether we are interested in the average or maximum secrecy setting. The definitions work analogously for both in those cases and when it is not clear from context we specify average or maximum secrecy in the index. We define the maximal achievable secrecy rate by

$$R^*(n, \varepsilon, \delta) := \max \left\{ \frac{\log(|\mathcal{M}|)}{n} : \exists (n, |\mathcal{M}|, \varepsilon, \delta) \text{ secrecy code} \right\} \ . \tag{2.24}$$

It makes intuitively sense that the security conditions in equations (2.22) and (2.23) make it hard for an attacker to obtain information about the message m. The following Theorem quantifies this statement for average secrecy.

Theorem 2.3 ([YSP16] Thm. 8)**.** *Let the output of an arbitrary list decoder $\mathcal{L}$ given Eve's observation E be denoted by $\mathcal{L}(E)$. Let δ be the secrecy parameter of the implemented secrecy code for the respective wiretap channel. Then it holds that the probability that the transmitted codeword is not in the output list of Eve's list decoder is lower bounded by*

$$P_{ME}(M \notin \mathcal{L}(E)) \geq 1 - \delta - \frac{L}{|\mathcal{M}|} \ . \tag{2.25}$$

Definition 2.19. *The secrecy capacity C_S of a wiretap channel is defined by*

$$C_S := \frac{1}{n} \limsup_{n\to\infty} R^*(n, \varepsilon, \delta) \tag{2.26}$$

for arbitrarily small values $\varepsilon > 0$ and $\delta > 0$.

The secrecy capacity for both channels is well known and given in the following Theorems.

Remark 2.2. *Notice that we did not distinguish the secrecy capacity for average and maximal secrecy because their value is the same. However for fixed n, ε, δ, the values $R^*_{avg}(n, \varepsilon, \delta)$ and $R^*_{max}(n, \varepsilon, \delta)$ can be different.*

Theorem 2.4 ([Wyn75])**.** *For the degraded wiretap channel with input A and outputs B and E for legitimate and eavesdropper, respectively, the secrecy capacity is equal to*

$$C_S = \max_{P_A} I(A;B) - I(A;E) \ . \tag{2.27}$$

Theorem 2.5 ([CK78])**.** *For the wiretap channel with input A and outputs B and E for legitimate and eavesdropper, respectively, the secrecy capacity is equal to*

$$C_S = \max_{P_{V,A}} I(V;B) - I(V;E) \ , \tag{2.28}$$

where V serves as an auxiliary random variable and it holds that $V \multimap A \multimap (B, E)$.

Remark 2.3. *Notice that in both cases the capacity does not depend on ε and δ. This changes for the task of determining $R^*(n, \varepsilon, \delta)$ for finite n. We introduce bounds on $R^*_{avg}(n, \varepsilon, \delta)$ and $R^*_{max}(n, \varepsilon, \delta)$ for finite n later in Section 2.3.2.*

2.2.2 Secret Sharing using Common Randomness

Secret sharing using common randomness has been investigated by Maurer in [Mau93] and by Ahlswede and Csiszàr in [AC93]. The problem is graphically illustrated in Fig. 2.2. In this section we explain the known results which will later be used in Chapter 4 in the context of Physical Unclonable Functions (PUFs). In the following we describe the problem of deriving a secret key that is shared between two terminals, in the following called Alice and Bob, when the terminals have access to common randomness and are able to use a public channel to send messages to each other. The source of common randomness is specified by a joint probability mass function P_{XYZ}. We denote the respective random variables specifying its outputs by X, Y, Z. This source is iid sampled n times and we denote the random variables specifying the output of this process by (X^n, Y^n, Z^n). The first terminal, in the following denoted by Alice, receives the sequence $X^n = (X_1, \ldots, X_n)$, while the second terminal (Bob)

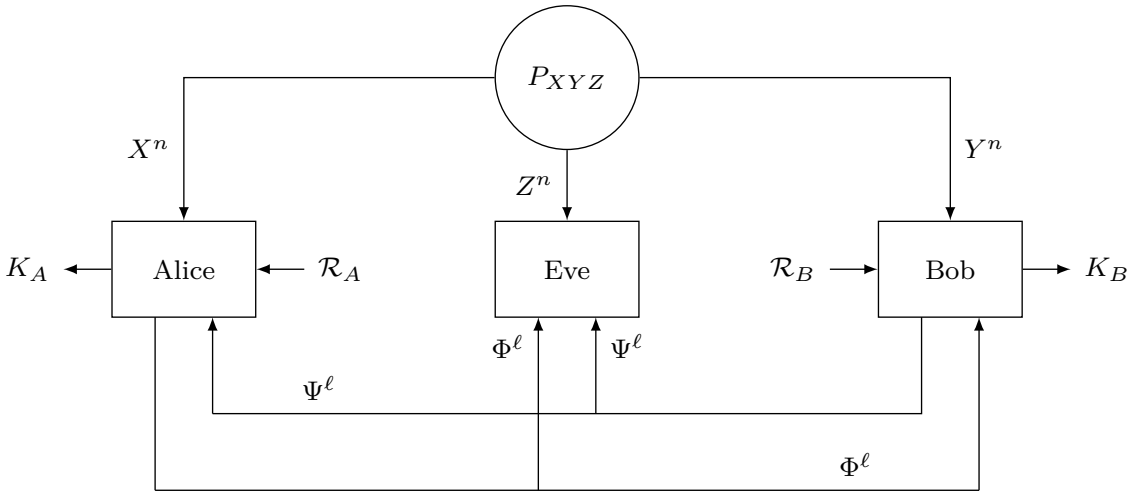

Figure 2.2: Secret Sharing using Common Randomness

gets the sequence $Y^n = (Y_1, \ldots, Y_n)$. The third terminal (Eve), which is an adversary trying to obtain knowledge about the secret key that Alice and Bob shall agree on, is provided with the sequence $Z^n = (Z_1, \ldots, Z_n)$. The probability mass function P_{XYZ} is publicly known, in particular by Alice, Bob and Eve. The goal of Alice and Bob is to reliably agree on a secret key while leaving Eve oblivious about it. To achieve this goal, Alice and Bob send messages to each other over the public channel that depend on their apriori knowledge of P_{XYZ} and their respective shares X^n or Y^n. Subsequent messages may also depend on previously received messages over the public channel coming from the other terminal. Eve is able to eavesdrop those messages but is unable to alter them or to insert additional messages into the public channel. We denote the i-th message sent from Alice to Bob by Φ_i and the i-th message sent from Bob to Alice by Ψ_i. Furthermore, we define $\Phi^i := (\Phi_1, \ldots, \Phi_i)$ and $\Psi^i := (\Psi_1, \ldots, \Psi_i)$. After Alice and Bob are finished with exchanging messages over the public channel, say after ℓ steps, Alice computes a key K_A and Bob computes a key K_B that are both within the same keyspace denoted by $\mathcal{K}$. For the case that both keys are equal we simply denote the key by K.

As for transmitting data securely over a wiretap channel, it is essential for Alice and Bob to have access to local randomness to randomize the encoding function for the messages sent over the public channel. Hence, we assume that Alice and Bob have access to local sources of randomness $\mathcal{R}_A$ and $\mathcal{R}_B$, respectively. In order to generate the messages to be transmitted over the public channel they make use of those such that it holds that

$$\Phi_1 = \Phi_1(\mathcal{R}_A, x^n), \quad \Psi_1 = \Psi_1(\mathcal{R}_B, y^n) \tag{2.29}$$

$$\Phi_i = \Phi_i(\mathcal{R}_A, x^n, \Psi^{i-1}), \quad \Psi_i = \Psi_i(\mathcal{R}_B, y^n, \Phi^{i-1}) \ , \tag{2.30}$$

where $\mathcal{R}_A$ and $\mathcal{R}_B$ are independent from the jointly distributed random variables X^n and Y^n corresponding the source of common randomness.

We next formalize the secret key rate which is the figure of merit that we would like to maximize for this problem.

Definition 2.20. *A secret key rate R is called **achievable** if for every $\varepsilon > 0$ and sufficiently large n there exists a secret key agreement scheme such that*

1. $Pr(K_A \neq K_B) < \varepsilon$
2. $I(Z^n, \Phi^\ell, \Psi^\ell; K) < \varepsilon$
3. $H(K) > R - \varepsilon$
4. $\log_2(|\mathcal{K}|) < H(K) + \varepsilon$.

We next give some interpretation to the properties that an achievable secret key rate has according to Definition 2.20.

The first property basically states that for sufficiently large n the probability that the key at the Alice terminal is unequal to the key at Bob's terminal is arbitrarily small. The second property states that no information can be deduced from the messages shared over the public channel and the source component Z^n about the key K. We recall at this point that random variables are stochastically independent if and only if their mutual information is zero and the second property says that we are able to approach this arbitrarily closely. The third property states that the entropy of K is basically at least R because ε is arbitrarily small. The fourth property states that the key is almost uniform over the keyspace $\mathcal{K}$.

The natural question of finding the maximal achievable secret key rate, in the following referred to as the *secret key capacity* as a function of P_{XYZ} has been answered in [AC93] and [Mau93].

Theorem 2.6. *The **secret key capacity** $\widetilde{C}_S$ denotes the maximal achievable secret key rate for a source of common randomness P_{XYZ} and is bounded by*

$$I(X;Y) - I(X;Z) \leq \widetilde{C}_S \leq I(X;Y|Z) \ . \tag{2.31}$$

Furthermore, the secret key capacity is achievable even if we only allow a single transmission over the public channel from Alice to Bob or from Bob to Alice.

Corollary 2.1 ([BB11], Corollary 4.1)**.** *If $X \multimap Y \multimap Z$ it holds that*

$$\widetilde{C}_S = I(X;Y) - I(X;Z) = I(X;Y|Z) \tag{2.32}$$

and hence upper and lower bound in (2.31) *are matching.*

Remark 2.4. *Notice that the secret key capacity C_S is very similar to the secrecy capacity of a degraded wiretap channel $P_{XYZ}(x,y,z) = P_X(x)P_{Y|X}(y|x)P_{Z|Y}(z|y)$ (see equation (2.27)) except for the fact that a maximization over the distribution P_X is omitted.*

Before we give a proof sketch for the achievability part of Theorem 2.6 we examine some special cases for P_{XYZ}. Let us assume that either $Z^n = X^n$ or $Z^n = Y^n$ holds. In this case the secret key capacity is zero. This is intuitively appealing as Eve is able to observe all communication over the public channel and one legitimate party has the same information from the source of common randomness as Eve. On the other hand if $P_{XYZ}(x,y,z) = P_{XY}(x,y)P_Z(z)$, observing Z^n gives no information about the shares X^n or Y^n. Hence, Z^n provides no useful information to Eve at all which is reflected by the fact that $I(X;Z) = 0$ in that case.

Remark 2.5. *In [AC93] it has first been discussed how to perform secret sharing with common randomness if the eavesdropper only has access to the messages transmitted over the public channel, i.e. the source was of the form P_{XY} and only later to examine the more general case (including Z). This is essentially equivalent to the case that Z is independent of X, Y. In this work, we chose the approach of directly introducing the model in the more general setting (including Z) as was also done in [Mau93].*

Proof sketch. In the following we sketch how the secret key rate given in Theorem 2.6 can be achieved using secrecy codes designed for secure data transmission over a (potentially degraded) wiretap channel. Furthermore, this construction only requires the transmission of a single block of length n from Alice to Bob over the public channel. We follow the structure of the proof given in [BB11, Chapter 4.2.1].

Let the secret key be encoded into a block of n symbols over $\mathcal{X}$, labelled by u^n. Alice would like to securely communicate the i-th symbol $u_i \in \mathcal{X}$ to Bob. The choice of this symbol is stochastically independent of the common randomness outputs X^n, Y^n, Z^n. She computes $u_i + x_i$ and sends the result over the public channel, where the addition is taken $\mod |\mathcal{X}|$. Bob receives his i-th dedicated symbol from the source of common randomness y_i and $u_i + x_i$ from the public channel. Eve receives z_i from the common randomness source and is able to eavesdrop $u_i + x_i$. This scenario can be interpreted as a wiretap channel. The channel's input is U, while the legitimate user's channel output is formed by the tuple $(U + X, Y)$ and the eavesdropper's output is formed by $(U + X, Z)$.

Assume that the sequence all possible sequences u^n are codewords of a wiretap code. It is possible to construct such a code by sampling the elements of all codewords from a single distribution P_U that can be chosen arbitrarily. According to the standard achievability proofs for secrecy codes of wiretap channels (see for instance [BB11, Chapter 3.4.1]) From Theorem 2.5 we know that a secrecy code for this channel with rate

$$R^* := I(U;Y,U+X) - I(U;Z,U+X) = H(U|Z,U+X) - H(U|Y,U+X) \quad (2.33)$$

is achievable by choosing the auxiliary random variable $V = U$. In case $X \multimap Y \multimap Z$ it follows that $U \multimap (U + X, Y) \multimap (U + X, Z)$ and by Theorem 2.4 that R^* is achievable.

For the choice of U being uniformly distributed over $\mathcal{X}$ and using properties of the one-time pad one is able to show that

$$R^* = I(X;Y) - I(X;Z) \ . \tag{2.34}$$

Hence, we have constructed a secret sharing scheme using common randomness from the source P_{XYZ} achieving the rate $I(X;Y) - I(X;Z)$. ■

The reason we provide this proof is that the methodology to perform secret sharing using common randomness using secrecy codes for wiretap channels naturally carries over to the finite blocklength regime. We come back to this point in Section 2.3.3.

2.3 Finite Blocklength Information Theory

In the previous sections we summarized results that make claims in the asymptotic setting as n goes to infinity. However, in practice we have to limit ourselves to some finite value for the blocklength n. For data transmission over point to point channels, natural questions to be asked are which rate R can be achieved for a given block length n and fixed block error probability P_e or which P_e can be achieved for fixed R and n.

In this section we introduce finite blocklength counterparts for block transmissions over discrete memoryless channels, wiretap channels in degraded and non-degraded form and for secret sharing using common randomness.

2.3.1 Discrete Memoryless Channels

Theorem 2.2 states that it only matters whether $R < C$ when $n \to \infty$ with respect to the achievable block error probability P_e. This arguably counterintuitive property occuring in the asymptotic setting does not hold in general in the case of finite blocklength coding.

An equivalent claim to Theorem 2.2 is that for the set of possible messages $\mathcal{M}$ and for arbitrarily small $\alpha > 0$ it holds that

$$|\mathcal{M}| = 2^{n(C-\alpha)-o(n)} \ . \tag{2.35}$$

For large enough n, $|\mathcal{M}| \approx 2^{nC}$ serves as a first order approximation. However, for moderate or even small values of n (typically less than 10000 but this number also depends on the channel) this approximation is rather inaccurate.

Hence, in order to analyze the maximal message size for blocklength n and block error probability ε, denoted by $M(n, \varepsilon)$, for a discrete memoryless channel $P_{Y|X}$ and

input distribution P_X the normal approximation has been proposed in [PPV10].

First, we define the information density which is closely related to the mutual information between two random variables X and Y.

Definition 2.21. *The information density $i(x;y)$ is defined by*

$$i(x;y) \coloneqq \log_2\left(\frac{\mathrm{d}P_{Y|X=x}}{\mathrm{d}P_Y}(y)\right) \tag{2.36}$$

where we used the Radon-Nikodyn derivative in the definition to cover both discrete and continuous random variables.

Theorem 2.7 (Normal approximation, [PPV10])**.** *For any input probability distribution P_X we have*

$$\log(M(n,\varepsilon)) = nI(X;Y) - \sqrt{nU(P,W)}Q^{-1}(\varepsilon) + \mathcal{O}(1) \ , \tag{2.37}$$

where Q^{-1} denotes the inverse of the Gaussian Q-function, $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z$ and the unconditional variance $U(P_X, P_{Y|X})$ is defined by

$$U(P_X, P_{Y|X}) = \mathrm{Var}(i(X;Y)) = \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}} P_X(x)P_{Y|X}(y|x)\log_2^2\left(\frac{P_{Y|X}(y|x)}{P_X(x)P_Y(y)}\right) - I(X;Y)^2 \ . \tag{2.38}$$

Next we give a well-known achievability result for discrete memoryless channels.

Theorem 2.8 (Random coding union (RCU) bound, [PPV10])**.** *For an arbitrary P_X there exists an (M,ε) code such that*

$$\varepsilon \leq \mathbb{E}[\min\{1, (M-1)P(i(\overline{X};Y) \geq i(X;Y)|X,Y)\}] \tag{2.39}$$

where $P_{XY\overline{X}}(a,b,c) = P_X(a)P_{Y|X}(b|a)P_X(c)$, i.e. the random variable X denotes the channel input, Y denotes the channel output and $\overline{X}$ denotes an independent random variable that has the same distribution as X.

The computation of the RCU-bound is difficult even for reasonably simple channels because one has to compute the expectations all possible sequences which is often infeasable. To circumvent this problem using the saddlepoint approximation has been introduced in [MF11; FSVVM+18]. For an introduction into saddlepoint approximation techniques we recommend [GC99].

We only give an overview on how to use the saddlepoint approximation to compute the RCU bound here and refer readers interested in the proofs on why this approximation is justified to the works mentioned above. However, we present the approximation used to obtain the results presented in Section 3.8.

Definition 2.22. *We define Gallager's $E_0(\rho, P_X)$ function to be*

$$E_0(\rho, P_X) := -\log_2 \left(\sum_{y\in\mathcal{Y}} \sum_{x\in\mathcal{X}} P_X(x) P_{Y|X}(y|x)^{\frac{1}{1+\rho}} \right)^{1+\rho} \quad . \tag{2.40}$$

Further we define $\hat{\rho} := \min(1, \hat{\rho}_0)$ with $\hat{\rho}_0$ as the root of

$$\frac{dE_0(\hat{\rho}, P_X)}{d\rho} = R = 1/n \log(M) \ , \tag{2.41}$$

$$W := R - \frac{dE_0(\hat{\rho}, P_X)}{d\rho} \tag{2.42}$$

and

$$V := -\frac{d^2 E_0(\hat{\rho}, P_X)}{d^2\rho} \ . \tag{2.43}$$

Theorem 2.9 ([MF11; FSVVM+18])**.** *The RCU bound of Theorem 2.8 can be approximated by*

$$\varepsilon \leq e^{n(\hat{\rho}R - E_0(\hat{\rho}, P_X))} \frac{1}{2} \left(\mathrm{erfcx}_1 \left(\hat{\rho}\sqrt{\frac{nV}{2}}, W\sqrt{\frac{n}{2V}} \right) + \mathrm{erfcx}_1 \left((1-\hat{\rho})\sqrt{\frac{nV}{2}}, -W\sqrt{\frac{n}{2V}} \right) \right) \ , \tag{2.44}$$

where

$$\mathrm{erfcx}_1(x, y) := \mathrm{erfc}(x - y) \exp(x^2 - 2xy) \tag{2.45}$$

using the saddlepoint approximation technique.

2.3.2 Degraded Wiretap and Wiretap Channels

As for DMCs the finite blocklength behaviour of degraded wiretap and wiretap channels have already been investigated [YSP16; YSP19]. In the finite blocklength regime the achievable rates depend on the concrete values for block error probability of the legitimate user ε and the security parameter δ. In contrast, similar to DMCs those parameters can be made arbitrarily small in the asymptotic regime as long as the rate is below the secrecy capacity.

Analogous to Section 2.3.1 the theorem dealing with the asymptotic behaviour (Theorem 2.4 and Theorem 2.5) merely provide the information that there exists a secrecy code of message cardinality

$$|\mathcal{M}| = 2^{nC_S + o(n)} \ . \tag{2.46}$$

C_S is only an approximation of $R^*(n, \varepsilon, \delta)$ and estimating $R^*(n, \varepsilon, \delta)$ by C_S is only reasonable for very large n. For small and moderate n it is essential to further analyze the $o(n)$ term in Equation (2.46). The following theorem established in [YSP19] gives

more precise upper and lower bounds on the exponent in Equation (2.46). The error term for these bounds is in the order of $\mathcal{O}(\log(n)/n)$.

Theorem 2.10 ([YSP19] Thm. 13)**.** *For a discrete memoryless (degraded or general) wiretap channel $P_{BE|A}$ with secrecy capacity C_S and for $\varepsilon + \delta < 1$ it holds that*

$$R^*_{max}(n, \varepsilon, \delta) \geq C_S - \sqrt{\frac{V_1}{n}} Q^{-1}(\varepsilon) - \sqrt{\frac{V_2}{n}} Q^{-1}(\delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \tag{2.47}$$

and

$$R^*_{avg}(n, \varepsilon, \delta) \leq C_S - \sqrt{\frac{V_c}{n}} Q^{-1}(\varepsilon + \delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \quad , \tag{2.48}$$

where

$$V_1 := \sum_{a \in \mathcal{A}} P_A(a) \left(\sum_{b \in \mathcal{B}} P_{B|A}(b|a) \log_2^2 \left(\frac{P_{B|A}(b|a)}{P_B(b)} \right) - D(P_{B|A=a} || P_B)^2 \right) \quad , \tag{2.49}$$

$$V_2 := \sum_{a \in \mathcal{A}} P_A(a) \left(\sum_{e \in \mathcal{E}} P_{E|A}(e|a) \log_2^2 \left(\frac{P_{E|A}(e|a)}{P_E(e)} \right) - D(P_{E|A=a} || P_E)^2 \right) \quad , \tag{2.50}$$

and

$$V_c := \sum_{a \in \mathcal{A}} P_A(a) \Bigg(\sum_{b \in \mathcal{B}, e \in \mathcal{E}} P_{BE|A}(b, e|a) \log^2 \left(\frac{P_{BE|A}(b, e|a)}{P_{E|A}(e|a) P_{B|E}(b|e)} \right) - D(P_{BE|A=a} || P_{B|E} P_{E|A=a})^2 \Bigg) \quad . \tag{2.51}$$

As in Theorem 2.7, Q^{-1} denotes the inverse of $Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \mathrm{d}z$.

2.3.3 Secret Sharing using Common Randomness

In this section we provide results for secret sharing using common randomness in the finite blocklength regime. This problem has been studied in [HTW16]. The authors established the secret key rate up to an error term in the order of $\mathcal{O}(\log(n)/n)$.

Theorem 2.11 ([HTW16],Thm.15)**.** *For every $\varepsilon, \delta > 0$ such that $\varepsilon + \delta < 1$ and iid (X^n, Y^n, Z^n) sampled according to joint pmf P_{XYZ} such that $X \multimap Y \multimap Z$, the maximal secret key rate for n, ε, δ is given by*

$$\widetilde{R}^*_{avg}(n, \varepsilon, \delta) = C_S - \sqrt{\frac{V'_c}{n}} Q^{-1}(\varepsilon + \delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \quad , \tag{2.52}$$

where

$$V_c' := \sum_{x\in\mathcal{X}} P_X(x)\left(\sum_{y\in\mathcal{Y},z\in\mathcal{Z}} P_{YZ|X}(y,z|x)\log_2^2\left(\frac{P_{YZ|X}(y,z|x)}{P_{Z|X}(z|x)P_{Y|Z}(y|z)}\right)\right. \\ \left. - D(P_{YZ|X=x}||P_{Y|Z}P_{Z|X=x})^2\right) . \tag{2.53}$$

Examining equation (2.52), we observe that its structure is similar to the bounds on the maximal secrecy rate in Theorem 2.10. The first term is the asymptotic result, i.e. the secret key capacity defined by the source of common randomness. This value is independent of n, ε, δ and only depends on P_{XYZ}. From this value another term that depends on P_{XYZ} via the dispersion coefficient V_c but also on n, ε, δ is subtracted. Notably this term decreases in n with speed $1/\sqrt{n}$ and hence Theorem 2.11 is consistent with the asymptotic result presented in Theorem 2.6. Finally there is an error term in the order of $\mathcal{O}(\log(n)/n)$ that goes down significantly quicker than $1/\sqrt{n}$ such that ignoring it approximates reality reasonably well for moderate values of n.

Remark 2.6. *The secret sharing protocol achieving the maximal secret key rate $\widetilde{R}^*_{avg}(n,\varepsilon,\delta)$ presented in [HTW16] requires two-way communication over the public channel. This is in contrast to the achievability proof in the asymptotic setting (Theorem 2.6) for which one-way communication suffices.*

The limitations of the achievability proof in [HTW16] has practical consequences, in particular in the context of applying the result to Physical Unclonable Functions which are our main concern in Chapter 4.

To this point, it is unclear to us whether the construction using wiretap coding is optimal in terms of maximizing the secret key rate in the finite blocklength regime. To the best of our knowledge it is even unknown whether it is possible to achieve the secret key rate in Theorem 2.11 using one-way communication over the public channel or not. For the interested reader, we refer to the discussion on the necessity of two-way communication to achieve the secret key rate determined by Theorem 2.11 in [HTW16, Section VII].

2.4 Background in Coding Theory

2.4.1 Motivation for Coding Theory

Theorem 2.2 specifies that there exists an encoder and a decoder such that the block error probability goes to zero as n goes to infinity. However, the proof of the theorem does not specify how to build encoders and especially decoders of sufficiently small complexity to be used in practice. Within the proof the codewords are generated at

random. Unfortunately, decoding a random linear code is an NP-complete problem [BMT78] and solving an NP-complete problem within the decoder is not feasible for many parameter sets in practice. In general we aim at having a decoding procedure having a complexity that is at least upper bounded by a polynomial of small degree with respect to the blocklength n.

There have been many attempts to find constructions for encoders and decoders of low complexity, notably Hamming codes [Ham50], Reed-Solomon codes [RS60], BCH codes [BRC60] and Reed-Muller codes [Mul54; Ree54]. All of these codes are of reasonably small complexity. However, it took until 1962, when Gallager presented low density parity check (LDPC) codes, the first capacity achieving code construction for the binary symmetric channel [Gal62] having a low complexity decoding algorithm. Even though Gallager's algorithm was of reasonably low complexity, at the time LDPC codes were invented it was not possible to manufacture chips capable of executing the decoding algorithm and knowledge about LDPC codes had been forgotten by most of the scientific community until the invention of Turbo codes [Ber95; BGT93]. This lead to the reinvention of LDPC codes [MN97] and due to the progress in semiconductor manufacturing over the past 30 years, it was possible to integrate encoders and decoders on a chip. Later, polar codes have been introduced by Stolte [Sto02] and Arıkan [Ari09]. Those works solved the problem of finding a code construction of low complexity achieving the capacity of any binary input discrete memoryless channel. In [TV15] it has been shown that small block error probabilities can be achieved by using list decoding and CRC precoding.

2.4.2 Limitations of LDPC and Polar Codes for Security Applications

LDPC and polar codes are both decoded iteratively and therefore calculating the block error probability for finite blocklengths is difficult. A standard way to estimate the block error probability is to use a Monte-Carlo simulation based approch. The idea is simply to simulate the channel and count how often the decoder fails to output the transmitted codeword to estimate the block error probability.

However, this methodology has the drawback that estimating very low ($< 10^{-9}$) block error probablities takes requires a huge amount of samples and is therefore computationally expensive. For estimating the error probability in most relevant communication scenarios a Monte Carlo based approach is usually sufficient because it is mostly unnecessary to achieve block error rates below 10^{-9}. In this work however, we are frequently aiming at block error rates that are approximately 2^{-128} to establish the security of public key cryptosystems and this cannot be simulated anymore with acceptable computational effort. For algebraic codes it is often possible to compute bounds on the block error probability given a certain channel model and simulations are not necessary. Therefore, for cryptographic purposes algebraic codes are often

preferable even though they might not be capacity achieving for the channel models under consideration.

2.4.3 Linear Codes

The definitions and statements in this sections are textbook knowledge and can be found in any textbook on coding theory, e.g. [Rot06; Bos13].

Definition 2.23. *We denote the Hamming weight $wt_H(\mathbf{v})$ of a vector $\mathbf{v} \in \mathbb{F}_q^n$ by the number of its nonzero entries. Furthermore, we denote the Hamming distance $d_H(\mathbf{v}_1, \mathbf{v}_2)$ between $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}_q^n$ by*

$$d_H(\mathbf{v}_1, \mathbf{v}_2) \coloneqq wt_H(\mathbf{v}_1 - \mathbf{v}_2) \ . \tag{2.54}$$

Definition 2.24. *A linear code $\mathcal{C}$ of length n with elements in $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$. Its minimum distance d is defined as*

$$d \coloneqq \min_{c_1, c_2 \in \mathcal{C}} d_H(c_1, c_2), \ \forall c_1 \neq c_2 \in \mathcal{C} \ . \tag{2.55}$$

We say that $\mathcal{C}$ is an $[n, k, d]_q$ code.

Proposition 2.1. *For the minimum distance d of a linear code $\mathcal{C}$ it holds that*

$$d \coloneqq \min_{c_1, c_2 \in \mathcal{C}} d_H(c_1, c_2) = \min_{c \in \mathcal{C}} wt_H(c) \tag{2.56}$$

because the difference between two codewords is again a codeword.

The interest of coding theorists in the minimum distance of a code stems from the following lemma.

Lemma 2.1. *A code of length n with minimum distance d can uniquely correct up to $\lfloor d-1 \rfloor/2$ errors or up to $d-1$ erasures.*

Next we introduce the *Gilbert–Varshamov* bound first shown in [Gil52; Var57] that guarantees the existence of an $[n, k, d]_q$ code.

Theorem 2.12 (Gilbert-Varshamov bound)**.** *Let n denote the blocklength, let k be the code dimension and let the minimum distance be denoted by d. If it holds that*

$$q^{n-k} > \sum_{i=0}^{d-2} \binom{n-1}{i} (q-1)^i \ , \tag{2.57}$$

then there exists an $[n, k, d]_q$ linear code over the field $\mathbb{F}_q$.

The bound in Theorem 2.12 does not tell how to obtain an efficiently decodable code having the parameters suggested by the Gilbert-Varshamov bound.

For this reason, next we consider specific classes of codes. Those codes also feature specific structure that enables efficient decoding at the receiver.

The definitions in Sections 2.4.4 and 2.4.5 have been taken from [WZPLW20]. They can also be found in textbooks, e.g. [Bos13; Rot06].

2.4.4 (Generalized) Reed Solomon Codes

Reed Solomon codes can be defined as evaluations of polynomials. They have been first introduced in [RS60].

Definition 2.25. *Let $\alpha_0, \alpha_1, \ldots, \alpha_{n-1} \in \mathbb{F}_q$ be non-zero distinct and let $\nu_0, \ldots, \nu_{n-1} \in \mathbb{F}_q$ be non-zero with $n \leq q-1$. Then a (Generalized) Reed Solomon Code (GRS code) of length n and dimension k can be defined by the set of vectors*

$$\{(\nu_0 u(\alpha_0), \nu_1 u(\alpha_1), \ldots, \nu_{n-1} u(\alpha_{n-1})) : u(x) \in \mathbb{F}_q[x] \text{ and } \deg u(x) < k\} \ . \tag{2.58}$$

Furthermore, for the minimum distance d it holds $d = n - k + 1$.

2.4.5 BCH Codes

The disadvantage of GRS codes is that their field size needs to be at least equal to the blocklength n. If codes of smaller alphabets are required it is possible to construct subfield-subcodes of GRS codes, also referred to as alternant codes. BCH codes and Goppa codes [Gop70] are prominent examples. In this section we introduce BCH codes. BCH codes have been introduced in [BRC60].

Next we briefly recap some fundamental properties of BCH codes. BCH codes are subfield subcodes of generalized Reed-Solomon codes. An interesting property they have is that we can use the fact that their minimum distance is at least as large as the minimum distance of the superseding GRS-code. Furthermore, in contrast to GRS-codes they can always be defined over the base field and thus they do not have the undesirable property of GRS codes that the field size cannot be smaller than the blocklength n.

Definition 2.26 (Cyclotomic cosets)**.** *Let n and q be positive integers with $gcd(n, q) = 1$ and let m be the smallest integer such that $n|(q^m - 1)$. Then the cyclotomic cosets C_i are defined by:*

$$C_i := \{i \cdot q^j \mod n, \forall j \in \{0, 1, \ldots, n_i - 1\}\} \ , \tag{2.59}$$

where n_i is the smallest integer such that $i \cdot q^{n_i} = i \mod n$.

Definition 2.27 (BCH code)**.** *Let $D = C_{i_1} \cup C_{i_2} \cup \cdots \cup C_{i_l}$ be the union of $l \geq 1$ distinct cyclotomic cosets with respect to n and $n|(q^s - 1)$ for some $s > 0$. Then let $\alpha \in \mathbb{F}_{q^s}$ be an element of order n. Then an $[n, k, d]_q$ BCH code is defined by*

$$g(x) = \prod_{i \in D} (x - \alpha^i) \ . \tag{2.60}$$

The codewords are then specified by the set

$$\mathcal{C} = \{u(x)g(x) : u(x) \in \mathbb{F}_q[x] \text{ and } \deg u(x) < k\} \ . \tag{2.61}$$

Lemma 2.2 (BCH bound)**.** *Let $\mathcal{C}$ be an $[n, k, d]_q$ BCH code where $n|(q^s-1)$ and α is an element of order n. If there exist integers b, δ with $\delta \geq 2$ and $\{b, b+1, \ldots, b+\delta-2\} \subseteq D$, it holds that $d \geq \delta$.*

2.5 Basics of Cryptography

2.5.1 Historical Development

In order to design secure communication infrastructure cryptography has been essential over the past decades. This discipline already dates back to the Roman empire when Julius Caesar made use of one of the first cryptosystems [Kah96]. Even though his scheme was far from what we consider secure with respect to modern security paradigms, it shows for how long human beings are already aiming to distribute messages amongst different parties in a secure manner. Unauthorized access to the message by an attacker reading the ciphertext was only prevented in case an attacker was not aware of the construction of the encryption algorithm. This is nowadays referred to as "security through obscurity".

Observing this tendency, Auguste Kerckhoffs [Pet83] stated that the security of a cryptographic scheme must only depend on the fact that the private key is kept secret and not on keeping the description of the cryptosystem itself secret.

Designing a scheme according to this principle has the advantage that the entire scientific community can scrutinize proposed algorithms and report vulnerabilities such that insecure schemes can hopefully be replaced before unauthorized malicious users exploit the vulnerability.

For hundreds of years researchers proposed different cryptographic schemes, claimed to be secure under the assumption that the private key was only known to the legitimate parties of the communication (the sender and the legitimate receivers). Frequently, attackers were still able to break those systems though. Schemes were considered secure by their users as long as no vulnerabilities have been reported. Famously, Turing was able to break an encryption machine named "Enigma" used throughout the second world war by the German Wehrmacht [Kah96]. The construction of encryp-

tion schemes by designers and cryptanalysts breaking them over the centuries rather recently lead to a more rigorous formalism in the design of algorithms. Proving the security of cryptosystems by reducing their security to hard mathematical problems became a popular approach.

2.5.2 Modern Cryptography

Shannon's work [Sha49] is widely considered to mark the start of the start of designing cryptographic schemes by using mathematically precise definitions and security proofs. In this work, he introduced the notion of perfect secrecy and proposed a cryptosystem meeting this notion which is nowadays often referred to as the one-time pad.

Definition 2.28 (perfect secrecy)**.** *Let C denote the random variable specifying the ciphertext and M denote the corresponding plaintext of an encryption scheme. The scheme is called perfectly secure if and only if*

$$P_{M|C}(m|c) = P_M(m) \quad \forall m \in \mathcal{M}, c \in \mathcal{C} \ , \tag{2.62}$$

where $\mathcal{M}$ denotes the set of possible plaintexts and $\mathcal{C}$ denotes the ciphertext space.

Definition 2.29 (One-time pad)**.** *The one-time pad is an encryption scheme that encrypt a message $m \in \mathbb{Z}_q^n$ using a uniformly sampled and pre-shared key $k \in \mathbb{Z}_q^n$ by computing the ciphertext*

$$c = m + k \mod q \ . \tag{2.63}$$

In order to decrypt the ciphertext c, the legitimate receiver uses the pre-shared key k and computes

$$m = c - k \mod q \ . \tag{2.64}$$

As shown in [Sha49] this scheme achieves perfect secrecy (Def. 2.28).

Definition 2.28 basically says that the message and the ciphertext are stochastically independent. In other words, an attacker observing the ciphertext c cannot use this knowledge to improve the output of an estimate for the plaintext m compared to the case where he did not observe c prior to outputting the estimate. Notice that perfect secrecy, frequently also called information theoretic security, does not impose any restrictions on the computational power of an attacker, i.e. it is implicitly assumed that an attacker has infinite computational power.

The one-time pad achieves the desirable notion of perfect secrecy but has the huge practical drawback that its key size has to be as large as the plaintext size used for performing encryption. This property not only holds for the one-time pad but as Shannon showed for any cryptosystem achieving perfect secrecy. So in that sense the one-time pad is optimal and cannot be improved.

Even though perfect secrecy is a desirable property, due to the aforementiond inherent limitations, other notions not suffering from those limitations have been proposed.

Designers of cryptographic schemes typically assume the attacker to be computationally bounded. Relaxing the security of the scheme to only resist computationally bounded attackers enables cryptographers to design encryption schemes with much smaller parameters.

So far we have given several milestones that mark attempts to secure communication between legitimate parties against adversaries. Now we treat matters in a more formal way. In order to prove the security of a cryptosystem, we have to specify what capabilities a potential attacker is assumed to have. Furthermore, within the proof all assumptions made by the designer have to be precisely stated. There shall be made as few and as weak assumptions as possible and it shall be investigated whether they are plausible and hold in practice during the investigation of the scheme's security.

In the following we will introduce several cryptographic notions. Those are nowadays textbook knowledge and are for instance described in [KL21]. This book also serves as the major reference for this section.

First we define the notion of a probabilistic polynomial time algorithm. When we speak loosely about efficient algorithms this is the class that we are usually referring to.

Definition 2.30 (PPT-algorithm)**.** *A probablistic polynomial time algorithm is an algorithm with input length parameter n, having access to a source of randomness whose runtime can be upper bounded by a polynomial $p(n)$.*

The notion of polynomial time algorithms is especially appealing because the composition of polynomials is always still a polynomial. That means that it does not depend on the instruction set of the machine on which an algorithm is running whether an algorithm runs in polynomial time or not.

Definition 2.31 (Negligible Function)**.** *A negligible function f from the natural to the non-negative real numbers in some parameter λ satisfies that there exists a number N such that for all $\lambda > N$ it holds that $f(\lambda) < 1/p(\lambda)$ for every positive polynomial p.*

Definition 2.31 basically states that the function f shall asymptotically decrease faster than the inverse of any polynomial function.

Definition 2.32. *The security parameter of a cryptographic scheme λ defines the permissible complexity of the best algorithm breaking the scheme.*

In the following we denote the set of possible keys by $\mathcal{K}$, the set of possible messages by $\mathcal{M}$ and the ciphertext space by $\mathcal{C}$.

Definition 2.33. *In order to define a private key encryption scheme Π we require three probabilistic polynomial time algorithms and define $\Pi := (Gen, Enc, Dec)$.*

1. *The key generation algorithm Gen takes as its input a security parameter λ and outputs the secret key $k \in \mathcal{K}$.*

2. *The encryption function takes a plaintext message m and the key k and outputs $c \coloneqq Enc(m, k)$.*

3. *The decryption function takes as its inputs a ciphertext c and the secret key k and outputs an estimate $\hat{m}$ for the message m, i.e. $\hat{m} = Dec(c, k)$.*

Definition 2.34. *We say that an encryption scheme is correct if it holds that*

$$Dec(Enc(m, k), k) = m \ . \tag{2.65}$$

In [GM84] Goldwasser and Micali defined the notion of semantic security. This is often seen as the computational equivalent of perfect secrecy. It basically states that an eavesdropper cannot obtain information about the plaintext by observing the ciphertext that he could not have obtained without the ciphertext.

Even though semantic security is intuitively appealing and captures the requirements we have on encryption schemes well, it is difficult to show that an encryption scheme fulfills this definition. Therefore, to prove security the following equivalent notion is often preferred. The basic idea is to formalize the security of encryption schemes in the form of a game for two players, an adversary $\mathcal{A}$ and a challenger. The adversary is choosing two messages $m_0, m_1 \in \mathcal{M}$ and interacts with the challenger who mimics an encryption oracle. By this procedure the adversary obtains a ciphertext corresponding either to an encryption of m_0 or m_1 and has to decide to which of the two plaintexts c corresponds. $\mathcal{A}$ wins this game if he outputs the correct index and loses otherwise. The scheme is secure if $\mathcal{A}$ cannot win this game with probability non-negligibly better than $1/2$. More formally:

Definition 2.35. *We define security against passive eavesdroppers by the following indistinguishability experiment $PrivK^{eav}_{\mathcal{A},\Pi}$.*

1. *The adversary $\mathcal{A}$ chooses two plaintexts $m_0, m_1 \in \mathcal{M}$ and sends it to the challenger.*

2. *The challenger uses the key generation to obtain a secret key sk.*

3. *The challenger samples a bit b uniformly at random, computes $c = Enc(m_b, sk)$ and sends it to $\mathcal{A}$.*

4. *$\mathcal{A}$ outputs a bit b' and sends it to the challenger.*

The adversary $\mathcal{A}$ wins this game if $b = b'$. We say that $\mathcal{A}$ achieves an advantage $Adv(PrivK^{eav}_{\mathcal{A},\Pi}) \coloneqq |Pr(b = b') - \frac{1}{2}|$ compared to random guessing. We say that a cryptographic scheme is secure against passive eavesdroppers if the best possible adversary $\mathcal{A}$ has only negligible advantage, i.e.

$$Adv(PrivK^{eav}_{\mathcal{A},\Pi}) < negl(\lambda) \tag{2.66}$$

with λ being the security parameter of the scheme.

Intuitively, security against passive eavesdroppers shall be the bare minimum that an encryption scheme has to achieve. However, practice has shown that this notion may be insufficient in many practical scenarios.

Definition 2.36. *An encryption scheme as said to be secure against chosen plaintext attacks if the advantage of an adversary $\mathcal{A}$ having additional access to an encryption oracle throughout the game described in Definition 2.35 is negligible in the security parameter.*

Remark 2.7. *Notice that the encryption oracle enables the adversary to compute both $Enc(m_0, k)$ and $Enc(m_1, k)$. Therefore, such an encryption cannot be deterministic as in this case the adversary could just compare the challenge c with the two encryptions.*

Definition 2.37. *An encryption scheme is said to be secure against chosen ciphertext attacks if the advantage of an adversary $\mathcal{A}$ having additional access to an encryption and a decryption oracle throughout the game described in Definition 2.35 is negligible in the security parameter λ with the only constraint that he is not allowed to use the challenge c as an input to the decryption oracle.*

For IND-CCA security it is distinguished whether access to the decryption oracle is only given before the adversary receives the challenge c or also afterwards. Throughout this work when we speak about IND-CCA security we mean the latter case. Throughout the literature this is also often referred to as IND-CCA2 security.

We have already stated that perfect secrecy is only achievable if the key is at least as long as the message to be encrypted. Modern cryptography takes the approach of showing that breaking a cryptosystem leads to an algorithm that enables the attacker to solve a well understood provably complex/hard mathematical problem. An algorithm efficiently solving the underlying mathematical problem an algorithm efficiently breaking the cryptosystem consequently cannot exist. A proof for the security of a cryptographic scheme is called a *proof by reduction*. Once such a proof has been accomplished the designer of the scheme chooses the parameters of the cryptosystem such that the best known attack has a complexity of 2^λ, where λ denotes the security level of the scheme. Provably complex problems are for example NP hard and NP complete problems. The assumption that needs to be made here is that the complexity classes P and NP are not equal. Proving or disproving this assumption is one of the famous Millenium problems. For more information on the Millenium's problems see for instance [CJW06].

2.5.3 Public Key Cryptography

So far we have only considered private key encryption schemes for which the key for encryption and decryption are the same. This implies that the communication parties

need to share this key via a secure channel, i.e. by meeting in person and exchanging a storage medium containing the key prior to using the encryption algorithm.

In many applications this procedure is undesirable and to circumvent the problem of secret key distribution public key cryptography has been invented. The main idea has already been found in 1974 by Ralph Merkle (publication in 1978 [Mer78]) which inspired Whitfield Diffie and Martin Hellman [DH76] in the construction of a scheme that securely establishes a secret key even though they do not have any pre-shared information and their entire communication can be eavesdropped by an attacker. The security of the Diffie-Hellman key exchange relies on the hardness of the discrete logarithm problem (DLP) in cyclic groups.

Definition 2.38. *A public key encryption scheme consists of the three PPT-algorithms* (Gen, Enc, Dec).

1. *The key generation algorithm Gen takes as its input the security parameter* λ *and outputs the private (secret) key sk and the public key pk.*
2. *The encryption function takes as its input a plaintext m and the public key pk and outputs* $c \coloneqq Enc(m, pk)$.
3. *The decryption function takes as its input a ciphertext c corresponding to a message m and the private key sk and computes* $\hat{m} = Dec(c, sk)$.

The correctness of the scheme is defined analogously to private key encryption schemes (Definition 2.34).

In 1978 [RSA78] Rivest, Shamir and Adleman proposed an encryption scheme with two keys, one of them being public and the other one being private for each communication participant. The public key is used within the encryption while the private key is used within the decryption algorithm.

RSA can be used to encrypt messages with another parties public that can then only be decrypted with the corresponding private key. Furthermore, it is possible to use the scheme to share the key of a private key encryption scheme over an insecure communication channel. The second application is perhaps the more common application of the scheme. Practice has shown that symmetric encryption schemes achieve better parameters and implementation complexity than asymmetric ones. The security of the RSA cryptosystem relies on the hardness of integer factorization.

2.6 Post-Quantum Cryptography

The security of most currently deployed asymmetric encryption schemes as well as digital signatures is based on the hardness of integer factorization or the discrete logarithm problem. In 1994, Shor developed quantum algorithms that are able to solve

both of these problems in polynomial time with respect to the size of the integer to factorize or the size of the group over which the discrete logarithm problem is defined [Sho94; Sho99]. Quantum computers with a sufficient amount of qubits to actually break schemes like RSA do not exist yet. However, driven by companies like Google and IBM significant progress has been made recently. Hence, it is essential to develop *post-quantum* (PQ)-secure cryptographic schemes due to the requirement of long-term security for devices that are hard to update (e.g., satellites). Furthermore, investigation of cryptographic schemes and the development of hardware and software implementations is a challenging task which requires time and effort of the cryptographic community. Another quantum algorithm for which there exists no efficient classical counterpart is the one by Grover [Gro96]. This one however only halfes the complexity exponent, i.e. brute-forcing an exponentially large space still takes exponential time. To take this possiblity for an attacker into account it suffices to increase the parameters of cryptographic schemes accordingly. For symmetric encryption schemes or hash functions this usually results in doubling the key sizes.

Due to the developments of Shor, the cryptographic community has been showing vast interest in finding cryptographic algorithms for which the security is based on the hardness of alternative mathematical problems other than integer factorization and the discrete logarithm problem. There were several approaches introduced or rediscovered. There are constructions on various hard mathematical problems. We mention algorithms based on hard problems in coding theory [McE78; ABC+22; MAB+18; ABB+22], on problems related to isogenies [DFKL+20], on the hardness of breaking hash functions [BHK+19] and on hard problems related to lattices [ABD+21; LLJ+17a; AAB+19; DKRV18; HPS98; BDK+18; FHK+18]. In this work we focus on lattice based key encapsulation mechanisms.

2.6.1 Lattices

This section gives a brief introduction into lattices and hard mathematical problems related to them. For a more elaborate discussion on lattice based cryptography we refer to [HPSS14, Chapter 7].

A lattice is a discrete subgroup of the n-dimensional Euclidean space $\mathbb{R}^n$ equipped with the standard Euclidean distance and the standard scalar product.

Definition 2.39. *Let $\mathbf{b}_1, \ldots, \mathbf{b}_m$ be a set of $m \leq n$ linearly independent vectors in $\mathbb{R}^n$. A lattice $\mathcal{L}$ specified by those vectors is defined by*

$$\mathcal{L} \coloneqq \left\{ \mathbf{v} \in \mathbb{R}^n : \mathbf{v} = \sum_{i=1}^{m} \alpha_i \mathbf{b}_i,\ \alpha_i \in \mathbb{Z} \right\} \ . \tag{2.67}$$

Notice that in contrast to a vector space over $\mathbb{R}^n$ the coefficients forming the linear combination specifying the elements in $\mathcal{L}$ are taken from the ring $\mathbb{Z}$ instead of the field $\mathbb{R}$.

There exist several computationally hard problems related to lattices. The following ones have been introduced in [Ajt96]. For an explanation on the complexity classes mentioned in the following see for instance [AB09].

Definition 2.40. *The shortest vector problem (SVP) is the problem of finding a nonzero vector of shortest Euclidean magnitude in a lattice $\mathcal{L}$ given a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$, i.e.*

$$\mathbf{s} \coloneqq \arg\min_{\mathbf{v}\in\mathcal{L}} \|\mathbf{v}\|_2 \ . \tag{2.68}$$

The shortest vector problem is NP-hard under randomized reductions [Ajt98], [GM02, Chapter 4].

Definition 2.41. *Let $\mathbf{v} \in \mathbb{R}^n \setminus \mathcal{L}$ and let a basis for $\mathcal{L}$ be given by $\{\mathbf{b}_1, \ldots, \mathbf{b}_m\}$. The closest vector problem (CVP) is defined by the task of finding the vector $\mathbf{c}$ that is closest in Euclidean distance to $\mathbf{v}$, i.e.*

$$\mathbf{c} \coloneqq \arg\min_{\mathbf{v}'\in\mathcal{L}} \|\mathbf{v} - \mathbf{v}'\|_2 \ . \tag{2.69}$$

The closest vector problem is NP-complete [GM02, Chapter 3]. Furthermore, it has been shown in [GMSS99] that approximating SVP is not significantly harder than approximating CVP.

Definition 2.42. *The shortest independent vector problem (SIVP) is the problem of finding the shortest basis of a lattice. We define the length of a basis by the Euclidean length of its longest vector.*

This problem is NP-hard even in the slightly relaxed form of asking for an independent set of lattice vectors that is longer than the shortest possible one by a constant factor $\alpha > 1$ [BS99].

In [AD97] Ajtai and Dwork proposed a cryptosystem whose security is based on the hardness of the unique-SVP which is closely related to the SVP (Definition 2.40). Furthermore, in this work it has been shown that for the underlying lattice problem there exists a so-called worst-case to average-case reduction. This means that the existence of an efficient algorithm for solving a random instance of the problem implies the existence of an efficient algorithm for solving the worst instance of the problem.

2.6.2 LWE, RLWE and MLWE Problem

Because reducing the security of cryptographic schemes directly to the lattice problems specified in Definitions 2.40,2.41 and 2.42 the Learning With Errors (LWE) problem has been introduced by Regev in [Reg05; Reg09]. Practice has shown that it is easier to design cryptosystems and to reduce the security to the hardness of LWE rather than to one of the aforementioned lattice problems. We define the LWE problem and

two of its variants; the Ring Learning with Errors (RLWE) and the Module Learning with Errors (MLWE) problem below.

Definition 2.43 (LWE problem)**.** *Consider a set of samples of the form*

$$(a_i, b_i = a_i s + e_i),\ i = 1, \ldots, m, \tag{2.70}$$

where the a_i are drawn uniformly from $\mathbb{Z}_q$ and s as well as e_i are sampled from χ_k. The decisional version of the LWE problem is defined as the task of distinguishing samples drawn from the distribution specified in (2.70) *from samples drawn from the uniform distribution on $\mathbb{Z}_q \times \mathbb{Z}_q$, where the problem shall be solved correctly with non-negligible advantage compared to random guessing, i.e. the advantage should be lower bounded by a function which is not negligible in the security parameter.*

Definition 2.44 (RLWE problem)**.** *Consider a set of samples of the form*

$$(a_i, b_i = a_i s + e_i),\ i = 1, \ldots, m, \tag{2.71}$$

where the a_i are drawn from the uniform distribution on $\mathcal{R}_q$ and s as well as the e_i are sampled from $\chi_k(\mathcal{R}_q)$. The decisional version of the RLWE problem is defined to be the task of distinguishing samples drawn from the distribution specified in (2.71) *from samples drawn from the uniform distribution on $\mathcal{R}_q \times \mathcal{R}_q$, where the problem shall be solved correctly with non-negligible advantage compared to random guessing, i.e. the advantage should be lower bounded by a function which is not negligible in the security parameter.*

Definition 2.45 (MLWE problem)**.** *Consider a set of samples of the form*

$$(\boldsymbol{a}_i, b_i = \boldsymbol{a}_i \boldsymbol{s} + e_i),\ i = 1, \ldots, m, \tag{2.72}$$

where the $\boldsymbol{a}_i$ are drawn from the uniform distribution on $\mathcal{R}_q^l$, s is sampled from $\chi_k(\mathcal{R}_q^l)$ and the e_i are sampled from $\chi_k(\mathcal{R}_q)$. The decisional MLWE problem is defined to be the task of distinguishing samples drawn from the distribution specified in (2.72) *from samples drawn from the uniform distribution on $\mathcal{R}_q^l \times \mathcal{R}_q$, where the problem shall be solved correctly with non-negligible advantage compared to random guessing, i.e. the advantage should be lower bounded by a function which is not negligible in the security parameter.*

The main contribution in Regev's work is a reduction of the hardness of LWE to the hardness of approximating SVP. Several cryptosystems are basing their security on the hardness of LWE, e.g. [BCD+16]. The complexity of encryption and decryption of those schemes can be improved for systems basing their security on the closely related but more structured *Ring Learning with Errors* (RLWE) [LPR10; LPR13] or Module Learning with Errors (MLWE) [BGV14] problems. Furthermore, the comparably large

key sizes of LWE-based schemes can considerably be shrunk for RLWE/MLWE-based schemes.

The hardness of RLWE can be reduced to the hardness of solving the (approximate) SIVP in a subclass of lattices, so-called ideal lattices [LPR13] and the hardness of MLWE can be reduced to the hardness of the (approximate) SIVP in module lattices. Computations in RLWE/MLWE based schemes can be implemented by choosing their parameters in a clever way such that it is possible to use the number theoretic transform (NTT) for the required polynomial multiplications [AB75; DCRVV15; GFS+12; LSSR+15; PG14].

2.7 Physical Unclonable Functions

The following section is based on the introduction of [MXP+21].

2.7.1 Introduction to PUFs

Secure key generation and storage are the foundation for secure cryptographic operations. As a wide range of integrated circuits does not have access to secure key storage, Physical Unclonable Functions (PUFs) evaluate physical properties of a circuit to derive a unique fingerprint and thus generate a device-specific cryptographic key. A PUF can be seen as the fingerprint of a physical object [Mae12; HYKD14], emanated from minuscule manufacturing variations that vary from object to object. This physical fingerprint can, hence, contribute to a hardware root of trust for security applications, such as authentication or identification of a device, and also the generation of a key for cryptographic primitives.

Commonly used solutions to generate a PUF are silicon PUFs such as SRAM, Arbiter or Ring Oscillator PUFs [HYKD14] that bury the secret deeply inside a chip to bring the security of the key storage to the same level as the remaining chip while remaining fully CMOS compatible. Those measures force attackers to use advanced failure analysis techniques like electro-optical probing [LTBS16] or focused ion beams [HBNS13] to successfully attack the system. The aforementioned attacks go beyond the typical attacker model applied for example in the IoT such that PUFs offer effective and cost-efficient solutions to increase teh security of embedded systems.

2.7.2 SRAM PUF

As an example we are now considering the SRAM PUF in more detail. SRAM is widely available on embedded devices and a popular PUF primitive [GKST07]. Upon power-up, the state of SRAM cells is not deterministically defined, however they will almost always start up in the same state. Due to manufacturing variations in the coupled inverters forming the SRAM cell, the state of a particular physical SRAM cell on a chip

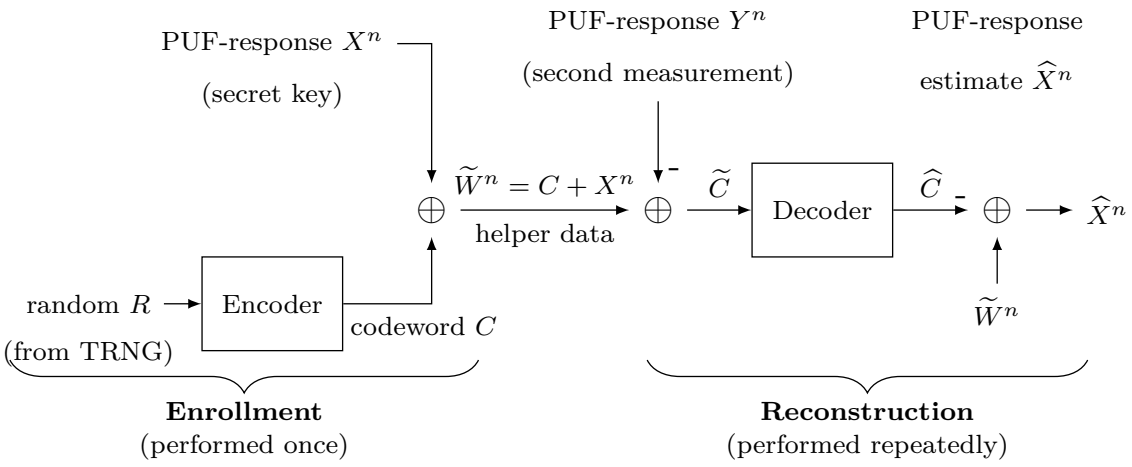

Figure 2.3: Simplified schematic of a key generation scheme based on a PUF.

will randomly be "0" or "1" but remain almost constant for each power-up. Multiple SRAM bits can therefore be grouped into a (randomly distributed) PUF-response.

Even though for a single device the content of the SRAM cells after power up is almost the same this behaviour is still insufficient. It is essential for cryptographic functions that any change at the input completely changes the output, hence it is essential that the reconstruction of the PUF values is precise.

2.7.3 Helper Data Algorithm

In the following we present a widely used method to integrate error correction capabilities into the generation of a cryptographic key using PUFs. We refer to this methodology as the helper data algorithm (HDA) [DRS04a]. A block diagram for the helper data algorithm is shown in Fig. 2.3.

The PUF response X^n, in this example the content of multiple SRAM cells after power up, is measured during the manufacturing process of the device, which is referred to as *enrollment*. Furthermore, during the enrollment a random number R is sampled from a True Random Number Generator (TRNG) which is used to select a codeword C at random from an ECC and the helper data $\widetilde{W}^n$ is computed according to $\widetilde{W}^n := C + X^n$. This helper data is published, e.g., in an external storage on the embedded system. Notice that the amount of SRAM cells corresponds to the length of the code in this case. We remark that additions and subtractions in this section are usually performed within the finite field over which the ECC is defined.

The goal of the helper data algorithm is to obtain the PUF response measured during the enrollment at another time when the PUF is measured again. We call this process *reconstruction*. The PUF measurement during the reconstruction phase is denoted by the random variable Y^n. Using the helper data $\widetilde{W}^n$ we compute $\widetilde{C} := \widetilde{W}^n - Y^n$.

Since Y^n does not have to be equal X^n even if the same device is used during the reconstruction phase we have that $\widetilde{C} = C + E^n$, where $E^n = X^n - Y^n$ denotes the error vector. If E^n is of sufficiently small weight, the codeword is correctly decoded. We denote the decoder's output by $\widehat{C}$. We then estimate the PUF response after applying error correction $\widehat{X}^n$ by computing $\widehat{X}^n = \widetilde{W}^n - \widehat{C}$.

The purpose of the helper data algorithm can be abstracted to the fact that it is not possible to perform coding over the channel from X^n to Y^n specified by the probability mass function $P_{Y^n|X^n}(y^n|x^n)$ as those values are the outputs of the PUF measurement without inherent structure. This channel is a function of the measurement noise and potential temperature dependence or aging effects. The helper data algorithm enables the integration of a structured ECC that can be chosen by the designer. This enables error correction within the reconstruction phase. Notice that the elements of the codeword C and the first estimate before decoding $\widetilde{C}$ are connected by virtually the same channel is C and X^n as well as $\widetilde{C}$ and Y^n differ only additively by the helper data $\widetilde{W}^n$. From the construction it also becomes obvious that the security level of the helper data scheme is upper bounded by the code dimension of the ECC in bits as this is simply the brute force complexity. The length of the code on the other hand determines the required length of the PUF-response and hence can be directly associated with the hardware complexity of the PUF, i.e. via the required number of SRAM cells for an SRAM PUF.

To remove any bias from the PUF-response, frequently the generated secret key is not equal to the PUF-response X^n but corresponds to the hash-value of X^n using a cryptographically secure hash-function. In general an approximate reconstruction of X^n is insufficient for key derivation because slight changes at the input of cryptographic functions, e.g. via the key, typically already lead to substantial changes at the function's output.

2.7.4 Relation to Secret Sharing Using Common Randomness

Recalling the secret sharing problem using common randomness presented in Subsection 2.2.2 and comparing it to the problem of reconstructing the value of a PUF during the reconstruction phase that has previously been measured throughout the enrollment phase, we observe that the problems are almost equivalent. This equivalence is outlined in the following.

The PUF responses during enrollment X^n and reconstruction Y^n can be interpreted as n samples from a source of common randomness specified by a joint distribution P_{XY}. Notice that in this model there is no adversarial output Z for the moment. The helper data for reconstruction $\widetilde{W}^n$ which is published by the helper data algorithm can be interpreted as data being sent over the public channel by the terminal having the enrollment data X^n (Alice in the secret sharing scenario). The reconstruction measurement data Y^n (Bob's share of the source of common randomness) is then used together with the message $\widetilde{W}^n$ received from the public channel. For a random code

with codewords that are iid sampled from the uniform distribution over the range of the RV X, the achievability proof sketch of Theorem 2.6 shows that the helper data scheme achieves the secret key capacity for the secret sharing problem using the resulting correlated source emanating from the PUF reconstruction problem. Notice that it is not possible to perform forward and backward transmissions over the public channel in this equivalence between secret sharing using common randomness and the PUF reconstruction problem. In the asymptotic setting it has been shown that one-way communication suffices to achieve the secret key capacity. In the finite blocklength regime this is unclear or even suggested not to hold (see Section 2.2.2). The construction achieving the secret key rate in Theorem 2.11 requires two-way communication over the public channel. Hence, it is unclear whether it is possible to generate a (possibly different) helper data scheme such that the rate in Equation (2.52) can be achieved for the coderate of the ECC. In fact, it is not obvious whether the helper data algorithm is optimal in maximizing the achievable rate if we restrict ourselves to one-way communication over the public channel. This question however is out of scope of this thesis.

2.7.5 Helper Data Algorithm for Analog PUFs

In contrast to a PUF outputting digital values like in the SRAM PUF, the output of analog PUFs cannot be directly used in the helper data algorithm introduced in Section 2.7.3. Therefore, during the enrollment phase the analog PUF output X^n needs to be quantized by an input quantizer and $S^n = Q(X^n)$. Furthermore, additional helper data is generated from X^n for the quantization required in the reconstruction phase. We denote this function by g and the resulting quantization helper data by W^n, in particular $W^n = g(X^n)$. A blockdiagram illustrating the additional quantization and helper data generation steps is given in Fig. 2.4. Input quantization and helper data generation are performed elementwise. With slight abuse of notation we sometimes also write $Q(X)$ for the quantization of a single PUF value throughout this thesis. $g(X)$ is treated the same way. We denote the codomain of Q by $\mathcal{S}^n$ and the codomain of g by $\mathcal{W}^n$. The quantization helper data W^n is published while S^n forms the key or is the preimage of the key via a cryptographically secure hash function and hence is kept secret. Like in the discrete case an ECC is fixed and a random number R sampled from a TRNG determines a codeword C. Similar to digital PUFs the helper data algorithm computes and publishes $\widetilde{W}^n = C + S^n$.

During the reconstruction phase the user measures the PUF again. We denote the outcome of this measurement by Y^n. From the quantization helper data W^n an output quantizer is derived which is then used to output an estimate $\widetilde{S}^n$ of S^n. The remaining steps are analogous to the discrete case. Basically the decoder is used to decode the erroneous codeword $\widetilde{C} = \widetilde{W}^n - \widetilde{S}^n$. Then the decoder outputs $\widehat{C}$ and $\widehat{S}^n = \widetilde{W}^n - \widehat{C}$ is computed. The key will be correctly recovered if $S^n = \widehat{S}^n$.

Since the quantization helper data W^n is derived directly from the PUF measure-

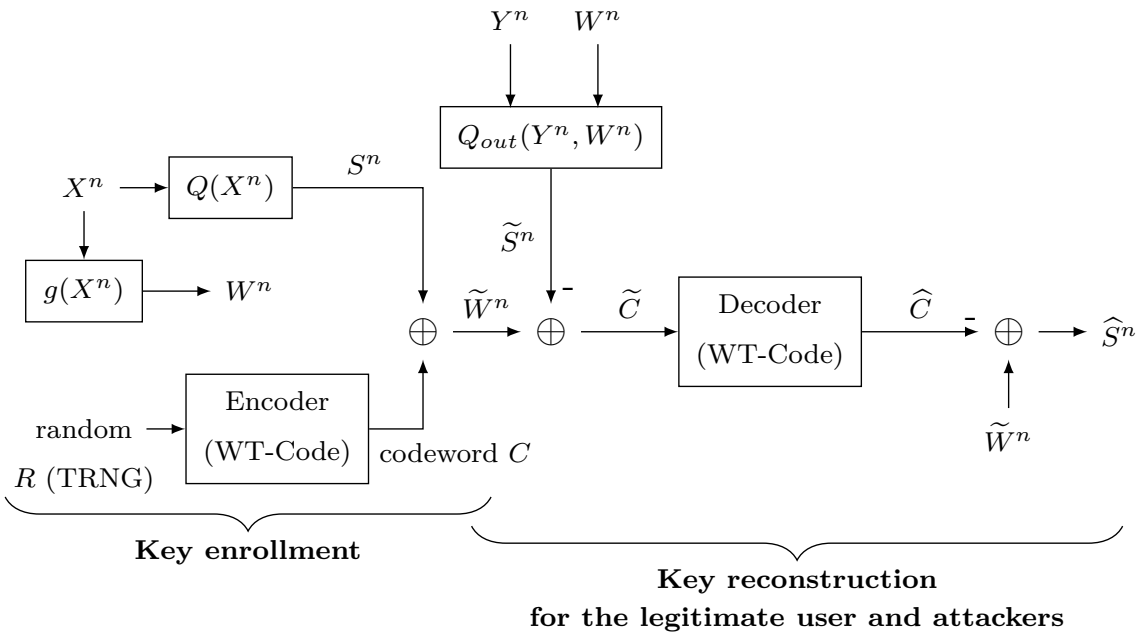

Figure 2.4: Key enrollment and key reconstruction for analog PUFs

ment during the enrollment phase X^n and so is the secret S^n, it has to be assured that from the public W^n it is impossible to derive information about S^n. The next section therefore deals with establishing quantization helper data that does not leak information about the secret S^n.

2.7.6 Zero Leakage Helper Data

Next we give the definition of zero leakage for helper data algorithms. This is the notion that we are aiming at for the generation of the quantization helper data in analog PUFs.

Definition 2.46. *A helper data algorithm is defined to have* zero leakage *if the PUF response (after quantization) S^n and the quantization helper data W^n are stochastically independent, i.e.*

$$P_{S^n|W^n}(s^n|w^n) = P_{S^n}(s^n),\ \forall s^n \in \mathcal{S}^n, w^n \in \mathcal{W}^n\ .$$

Sufficient and necessary conditions for helper data featuring zero information leakage about the secret are given in [GSVL16]. In this work it was shown that it is possible to construct a zero leakage helper data scheme using a function g on the PUF response to generate the helper data scheme, having the following properties:

1. g is strictly monotonous function (and therefore an injective) function from each quantization interval to the domain of the helper data $\mathcal{W}$.

2. Any other function g^* generating the helper data cannot lead to a better reconstruction performance or does not have the zero leakage property.

The main result that we are using from [GSVL16] is given below:

Theorem 2.13 (Thm. 4.8 [GSVL16])**.** *Let g be monotonously increasing on each quantization interval A_t, with $g(A_0) = \ldots = g(A_{N-1}) = \mathcal{W}$, where N denotes the number of quantization levels. Let x_t and x_u be from different intervals with $g(x_t) = g(x_u)$. Then in order to satisfy zero leakage the following condition is sufficient and necessary:*

$$\frac{F_X(x_t) - F_X(q_t)}{p_t} = \frac{F_X(x_u) - F_X(q_u)}{p_u} \ , \tag{2.73}$$

where q_t denotes the left border of the interval A_t and p_t denotes the probability that X is sampled to be in A_t, i.e. $p_t = Pr(X \in A_t)$. Analogous statements hold for q_u, A_u and p_u.

We refer to points x_t, x_u from different quantization intervals leading to the same helper data as *sibling points*. As mentioned in [GSVL16] Theorem 2.13 also leads to a natural way of computing helper data via

$$w = g(x_{t,w}) := \frac{F_X(x_{t,w}) - F_X(q_t)}{p_t} \ , \tag{2.74}$$

where $x_{t,w}$ denotes the point within the interval A_t leading to a helper data value of w. This is not the only optimal way to define the helper data but there is no way that leads to better performance during reconstruction while keeping the zero leakage property. Therefore, we take this approach for computing helper data throughout this work.

Lemma 2.3 ([SAS17] Lem. 1)**.** *The distribution of the helper data defined in equation* (2.74) *is the uniform distribution over the interval* $[0, 1]$*, i.e.*

$$f_W(w) = \begin{cases} 1 & \text{for } w \in [0,1] \\ 0 & \text{otherwise} \ . \end{cases} \tag{2.75}$$

Note that this construction facilitates to use the previously discussed equiprobable and equidistant input quantizations into zero leakage helper data algorithms.

In order to obtain the estimate $\widetilde{S}$ from the measurement during reconstruction Y the helper data is used to generate another quantizer. Its interval borders depend on W. For an equiprobable quantization, meaning that the input quantizer is formed such that S is uniformly distributed over its range, the construction of the output quantizer is also given in [GSVL16].

The more general case for arbitrary input distributions has been investigated in [SAS17]. The authors present an output quantizer for zero leakage helper data that

we are also using in the following. The computation of the interval borders of this quantizer is given in Theorem 2.14. We refer to Remark 2.9 for a more specific statement on what is meant by $p_{t-1} \not\ll p_t$.

Theorem 2.14 ([SAS17] Thm. 1)**.** *Let the values $\tau_0, \ldots, \tau_N$ denote the interval borders of the output quantizer used to obtain $\widetilde{S}$ from Y and let $p_{t-1} \not\ll p_t$. Let $\tau_0 = -\infty$ and $\tau_N = \infty$. Let $g_t^{-1}(w)$ be the unique value x in quantization interval t such that $g(x) = w$. Then choosing τ_t iteratively starting from following equation* (2.76) *gives the best reconstruction estimate for zero leakage helper data.*

$$\tau_s = \frac{\ln\left(\frac{p_{t-1}}{p_t}\right)}{g_t^{-1}(w) - g_{t-1}^{-1}(w)}\sigma_N^2 + \frac{g_{t-1}^{-1}(w) + g_t^{-1}(w)}{2} \tag{2.76}$$

Remark 2.8. *Notice that the choice for the output intervals given in Theorem 2.14 is consistent with Theorem 5.2 in [GSVL16] for uniform S.*

Remark 2.9. *If $p_s \ll p_{s-1}$ holds the symbol $s \in \mathcal{S}$ may be suboptimal irrespective of the channel output. This happens because the a-priori probability p_s of the symbol s is so large that for the equality point τ_s splitting the decision regions between the symbols $s-1$ and s it holds that $\tau_s < \tau_{s-1}$. If this happens the output quantizer has only $|\mathcal{S}|-1$ symbols and we compute*

$$\tau^* = \frac{\ln\left(\frac{p_{s-2}}{p_s}\right)}{g_s^{-1}(w) - g_{s-1}^{-1}(w)}\sigma_N^2 + \frac{g_{s-1}^{-1}(w) + g_s^{-1}(w)}{2} \quad . \tag{2.77}$$

In case $\tau^* < \tau_{s-2}$ we repeat the procedure. In this case the quantizer has only $|\mathcal{S}| - 2$ symbols.

The computation of the output intervals can either be done during the reconstruction phase or within the enrolment phase. In the latter case the helper data is not a single number $w \in [0, 1)$ but rather the set of all interval borders τ_s. This means that one can trade computational complexity during the reconstruction phase against storage consumption within the device, where the helper data needs to be stored.

Remark 2.10. *Observe that the reconstruction quantizer depends on the distribution of S. This distribution in turn depends on the input quantizer and since $\widetilde{S}$ depends on the reconstruction quantizer we have the peculiar case that the conditional pmf $P_{\widetilde{S}|S,W}$ in general depends on the input distribution P_S.*

3

Information and Coding Theoretic Analysis of Learning With Errors Based Cryptosystems

Abstract

Several cryptosystems based on the *Learning with Errors* (LWE) problem and its variants have been proposed within the NIST post-quantum cryptography standardization process, e.g., Kyber, Newhope, Frodo, LAC. All previously mentioned schemes have the drawback of a non-zero decryption failure rate (DFR). The combination of encryption and decryption for these kinds of algorithms can be interpreted as data transmission over a noisy channel. In this chapter we analyze the capacity of this channel. We show how to modify the encryption schemes such that the input alphabets of the corresponding channels are increased. In particular, we present lower bounds on their capacities which show that the transmission rate can be significantly increased compared to standard proposals in the literature. Furthermore, we investigate the problem of reducing the size of the ciphertext for a fixed number of plaintextbits, e.g., 256 bits for an AES key. We present results for the parameter sets of several cryptosystems. We use finite blocklength information theoretic concepts and BCH codes to obtain lower bounds on the plaintext bits per ciphertext bit that can be securely transmitted over the channel in case only one LWE block is considered.

The lower bound on the channel capacity of the channel resulting from the concatenation of encryption and decryption presented in this work has been published in the Transactions on Information Forensics and Security [MPWZ22]. This work deals with the maximization of the plaintextbits per ciphertextbit or similarly with the maximization of the asymptotic rate of the LWE channel. Parts of this work furthermore have been published in the proceedings of the 2020 Information Theory Workshop (ITW) [MPWZ21]. The results on minimizing the ciphertext and keysizes while maintaining the security level for the secure transmission of 256 bit have been accepted for the 2024 IEEE Information Theory Workshop (ITW).

3.1 Introduction

In 1994, Shor showed that cryptographic schemes that base their security on the hardness of integer factorization and the discrete logarithm problem are vulnerable against attackers having access to large scale quantum computers [Sho94; Sho99]. Due to this vulnerability, in 2016 at the PQCrypto conference, the National Institute of Standards and Technology (NIST) announced a competition for which cryptographers all over the world were invited to submit key encapsulation mechanisms (KEMs) and digital signatures that withstand quantum attackers [Moo20; Nat].

Several algorithms proposed within this standardization competition base their security on LWE or its variants RLWE and MLWE. This chapter investigates ways to improve the parameters of these schemes without reducing their security level.

It is possible to choose the parameter sets of LWE/RLWE/MLWE based public key encryption schemes such that the decryption of the ciphertext never fails, i.e. the concatenation of encryption a message with the public key and subsequent decryption using the private key outputs the original message. Although this property is desirable, there are practical reasons why a non-zero (but very small) *decryption failure rate* (DFR) is permitted by several algorithms (e.g., 2^{-174} for the Kyber1024 parameter set). It significantly reduces key sizes, the size of the ciphertext (for the same message length) and the complexity of the encryption and decryption algorithms. A low decryption failure rate is not only essential since retransmissions cost data rate but they also provide information for an attacker that tries to break cryptosystems [JJ00; HGNP+03; GN07; Flu16]. A possible measure to decrease the DFR is to use suitable *error-correcting codes* (ECC).

In [LKN+19] it was suggested to interpret the LWE-based cryptographic scheme Frodo [BCD+16] as a digital communication system. Exchanging messages between two parties in a secure manner using LWE/RLWE/MLWE based algorithms can also be considered as data transmission over a noisy channel, in the following referred to as the *LWE/RLWE/MLWE channel*. Consequently, we can find the channel capacity of this cryptographic channel by using Shannon's noisy channel capacity theorem presented in [Sha48]. To our knowledge, our works [MPWZ21; MPWZ22] are the first that analyze the capacity and other information-theoretic properties of this channel. In [RTA18] polar coded LWE-based symmetric key encryption schemes as well as wiretap coded LWE-based encryption have been investigated, while in this work we analyze LWE/RLWE/MLWE-based public key encryption schemes.

For an analysis on the impact of ECCs in NewHope Simple [ADPS16] see [FPS18]. In their work the effect of using a BCH code, an LDPC code and their concatenation is analyzed. However, only the influence of one specific BCH code is analyzed whereas in this work we optimize the BCH parameters with respect to different alphabet sizes. The BCH code construction presented in [MPWZ22] that is also shown in this thesis has later been improved by adding Gray Coding and vector quantization [BHK+22].

The analysis presented in this work provides a framework and can be applied to

various LWE/RLWE/MLWE based schemes. The objective of this framework is either to maximize the achievable rate of the channel or equivalently to maximize the amount of plaintext by a fixed amount of ciphertext bits or to minimize the ciphertext size for a fixed amount of plaintext bits as well as minimizing the sizes of private and public key. Maximizing the amount of plaintext per ciphertext bit especially makes sense if the public key encryption schemes shall be used to transmit large amounts of data over a public channel. The more common scenario however is to use public key cryptography to share a private key of a symmetric encryption scheme, e.g., AES [DR98], and then to encrypt the data transfer between the communication parties using symmetric encryption before sending it over the public channel. In this work we consider that the KEM is used to share a 256 bit key. We present asymptotic and finite length results utilizing the normal approximation and the RCU bound [PPV10] showing that ciphertext and key sizes can be significantly reduced.

For the analysis we chose to consider Kyber [ABD+21; BDK+18] and NewHope [AAB+19] within the main part of this chapter and for some results only Kyber. The reason for these choices is that we consider Kyber the most relevant scheme as it has been selected for standardization whereas NewHope has already been practically examined by Google as a candidate to achieve post quantum secure communication [Bra16]. Furthermore, we provide some results for Frodo [BCD+16; ABD+20] and LAC [LLJ+17a] in Appendix 3.C to show that it the framework can easily be applied to other relevant schemes. The two other lattice-based schemes NTRU [HPS98; CDH+] and Saber [DKRV18; BMD+] are both still part in the Round 3 of the NIST PQC. However, our framework can only be applied to schemes that with non-zero DFR. Therefore, we did not consider NTRU as the parameter choices for this scheme are designed such that decryption failures are impossible. Saber is a Learning with Rounding (LWR)-based scheme and the noise creation is therefore significantly different from the schemes considered in this work.

3.1.1 Outline

In Section 3.2 the general structure for many RLWE/MLWE-based public key encryption schemes is presented. In Section 3.3, we show how to connect RLWE/MLWE-based cryptosystems to communication theory in the Shannon sense. The consequences of these results naturally leads to the information-theoretic analysis presented in Section 3.4. In this section we also show how to bound the decryption failure rate of RLWE/MLWE-based schemes under the assumption of stochastically independent coefficient failures. In Section 3.5 we show the results for maximizing the achievable rate obtained by the previously introduced lower bound while in Section 3.6 we show the asymptotic results for the problem of transmitting 256 bit per MLWE block in Kyber. Section 3.7 deals with maximizing the achievable rates of the considered schemes and with the minimization of the decryption failure for fixed minimal bitrates using practical ECCs. In Section 3.8 we present finite blocklength achievability bounds on

reducing the ciphertext size of Kyber using the normal approximation as well as the RCU bound. Finally in Section 3.9 we sum up the results and conclude the chapter.

3.2 RLWE/MLWE-Based Cryptography

3.2.1 Ciphertext Compression and Decompression

Within the MLWE/RLWE based cryptographic schemes coefficients of polynomials are frequently compressed to reduce the size of the generated ciphertexts. At the receiver a decompression function is applied. Since the compression is lossy, the concatenation of compression and decompression only approximates its input.

We denote the output of the compression function on input z by z'. The compression function compresses each coefficient down to d_c bits and is defined by

$$z' = \mathsf{comp}_q(z, d_c) = \left\lceil \frac{z \cdot 2^{d_c}}{q} \right\rfloor \mod 2^{d_c} \ . \tag{3.1}$$

We denote the output of the decompression function on input z' by z''. The decompression function is defined by

$$z'' = \mathsf{decomp}_q(z', d_c) = \left\lceil \frac{z' \cdot q}{2^{d_c}} \right\rfloor \ . \tag{3.2}$$

The inputs to comp_q and decomp_q are to be represented in the range $\{0, \ldots, q-1\}$. We define both functions also for vectors of polynomials in $\mathcal{R}_q$ by applying them separately to each polynomial. The exact specification of the ciphertext compression within NewHope and Kyber can be found in [ADPS16] and [ABD+21], respectively.

3.2.2 Public Key Encryption Based on MLWE/RLWE

We consider the scenario that Alice would like to transmit a message to Bob using a public key encryption scheme. In order to do this, Bob generates a key pair (pk, sk) consisting of a public key pk and a secret key sk. The public key is then used by Alice to encrypt a message m to obtain a ciphertext c which she sends to Bob. Bob then uses his secret key (private key) sk and the ciphertext c to obtain an estimate (remember decryption may fail with very small probability) for the message m. Each *public key encryption* (PKE) scheme is composed of three functions: key generation, encryption, and decryption. The first one generates the required public and private keys, the second one is for encryption and the last one is for decryption. The basic building blocks for RLWE/MLWE-based schemes are presented in Algorithms 1, 2 and 3.

Algorithm 1: Key Generation

Input: n, q, k, l

$\boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{l \times l}$

$\boldsymbol{s}, \boldsymbol{e} \xleftarrow{\$} \chi_k(\mathcal{R}_q^l)$

$\boldsymbol{b} \leftarrow \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$

Result: $pk = (\boldsymbol{A}, \boldsymbol{b})$, $sk = \boldsymbol{s}$

Algorithm 2: Encryption

Input: $pk = (\boldsymbol{A}, \boldsymbol{b})$, $m \in \mathcal{M}$, (n, q, k, l), d_u, d_v

$\boldsymbol{s'}, \boldsymbol{e'} \xleftarrow{\$} \chi_k(\mathcal{R}_q^l)$

$e'' \xleftarrow{\$} \chi_k(\mathcal{R}_q)$

$\boldsymbol{u} \leftarrow \boldsymbol{A}^T\boldsymbol{s'} + \boldsymbol{e'}$

$v \leftarrow \boldsymbol{b}^T\boldsymbol{s'} + e'' + \mathsf{Encode/Map}(m)$

$\boldsymbol{u'} \leftarrow \mathsf{comp}_q(\boldsymbol{u}, d_u)$

$v' \leftarrow \mathsf{comp}_q(v, d_v)$

Result: $c = (\boldsymbol{u'}, v')$

Algorithm 3: Decryption

Input: $c = (\boldsymbol{u'}, v')$, $sk = \boldsymbol{s}$, (n, q, k, l), d_u, d_v

$\boldsymbol{u''} \leftarrow \mathsf{decomp}_q(\boldsymbol{u'}, d_u)$

$v'' \leftarrow \mathsf{decomp}_q(v', d_v)$

$\hat{m} \leftarrow \mathsf{Demap/Decode}(v'' - \boldsymbol{s}^T\boldsymbol{u''})$

Result: $\hat{m}$

Within RLWE/MLWE based cryptosystems the parameter n denotes the number of coefficients within the polynomials of the ring $\mathcal{R}_q$, q denotes coefficient modulus of $\mathcal{R}_q$, k parametrizes the error distribution (e.g., controls its variance) and l specifies the dimension of the matrices and vectors used within the algorithms. For RLWE based schemes it holds that $l = 1$. Additionally, in some RLWE-based (e.g. [AAB+19]) or Module-LWE based schemes (e.g. [ABD+21]) a technique called ciphertext compression is used to reduce the size of the ciphertext. In comparison to schemes which are based on the Learning with Rounding problem [BPR12] (e.g., [DKRV18], [BGML+18]), ciphertext compression for RLWE or MLWE-based schemes plays only a secondary role concerning the security of the encryption schemes.

3.2.3 Key Generation, Encryption and Decryption

For the following description of key generation, encryption and decryption it is valid to consider RLWE-based scheme as MLWE-based scheme with parameter $l = 1$. The random elements sampled in the presented algorithms are either sampled from the uniform distribution or from the error distribution.

The key generation for RLWE/MLWE based schemes is shown in Algorithm 1. First,

a matrix $\boldsymbol{A}$ is sampled uniformly from $\mathcal{R}_q^{l\times l}$. Then the vectors $\boldsymbol{s}$ and $\boldsymbol{e}$ are sampled from $\chi_k(\mathcal{R}_q^l)$ to compute $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{e}$. The public key is defined to be $pk = (\boldsymbol{A}, \boldsymbol{b})$ and the private key is defined to be $sk = \boldsymbol{s}$.

We denote the set of possible messages by $\mathcal{M}$. The encryption for RLWE/MLWE based schemes is shown in Algorithm 2. It involves apart from sampling polynomials according to predefined distributions and simple algebraic operations only ciphertext compression and the encoding and mapping of the message $m \in \mathcal{M}$ via the function $\mathsf{Encode/Map}$. To encrypt m, we first need to transform it into a polynomial in $\mathcal{R}_q$. We have a certain flexibility in the choice of the encoding and decoding functions which can be used to reduce the overall decryption failure probability of the scheme. Within the encoding step several algorithms utilize error-correcting codes (ECCs) to reach the required DFR of the scheme, thereby achieving the desired security level. For example BCH codes are deployed in LAC [LLJ+17a] and NewHope [AAB+19] uses a repetition code of length 4. Thus, a message that can be represented by k bits is encoded by an ECC of length at most n, with n being the number of coefficients of a polynomial in $\mathcal{R}_q$. Commonly the mapper takes the codeword and converts it into a polynomial in $\mathcal{R}_q$ by multiplying each bit of the codeword by $\lfloor q/2 \rfloor$ using the resulting sequence as the coefficients of the polynomial in sequential order.

The resulting ciphertext c consists of the tuple $(\boldsymbol{u'}, v') \in \mathcal{R}_q^l \times \mathcal{R}_q$ and is computed according to Algorithm 2. The steps to obtain them only involve sampling from χ_k and some finite field arithmetic in $\mathcal{R}_q$. The decryption of RLWE/MLWE based schemes is depicted in Algorithm 3. Bob first decompresses the ciphertext using decomp_q and then computes

$$\begin{aligned} v'' - \boldsymbol{s}^T\boldsymbol{u''} &= v + c_{N_v} - \boldsymbol{s}^T(\boldsymbol{u} + \boldsymbol{c_{N_u}}) \\ &= \boldsymbol{e}^T\boldsymbol{s'} + e'' - \boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c_{N_u}}) \\ &\quad + c_{N_v} + \mathsf{Encode/Map}(m) \end{aligned} \tag{3.3}$$

using his private key $\boldsymbol{s}$. We define the compression noise terms $\boldsymbol{c_{N_u}} := \boldsymbol{u''} - \boldsymbol{u}$ and similarly $c_{N_v} := v'' - v$. We split the result in equation (3.3) into two components: $\mathsf{Encode/Map}(m)$ and $\boldsymbol{e}^T\boldsymbol{s'} + e'' - \boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c_{N_u}}) + c_{N_v}$, where the latter can be interpreted as a noise term composed of terms sampled from the error distribution and compression noise terms.

Since all polynomials occurring within the noise are either caused by ciphertext compression or sampled from χ_k, it is likely that the coefficients of the noise are small in magnitude as long as the compression is not too strong. In order to decrypt the ciphertext, the quantity $v'' - \boldsymbol{s}^T\boldsymbol{u''}$ is computed and used as the demapper's input. Arguably the simplest demapping strategy was chosen for LAC. For this scheme the demapper examines whether the i-th coefficient $(v'' - \boldsymbol{s}^T\boldsymbol{u''})_i$ is closer to 0 or to $\lfloor q/2 \rfloor$ modulo q. If the demapper's input is closer to $\lfloor q/2 \rfloor$ for the respective coefficient then the demapper outputs 1 for the respective index i in the binary output vector

$\boldsymbol{d} = (d_1, \ldots, d_n) \in \mathbb{Z}_2^n$, otherwise it outputs 0.

$$d_i = \begin{cases} 0 & \text{if } |(v'' - \boldsymbol{s}^T \boldsymbol{u''})_i| \leq \left\lfloor \frac{q}{4} \right\rfloor \\ 1 & \text{otherwise} \end{cases}$$

In that sense we have defined a hard decision strategy.

Definition 3.1. *We refer to the event that $d_i \neq (\mathsf{Encode}(m))_i$ as a coefficient failure in the decoding procedure. We denote it by $\mathcal{E}$ and use indices in case we specify the respective coefficient.*

To reach the required decryption failure rate the vector $\boldsymbol{d}$ is put into the decoder afterwards (e.g. BCH decoder for LAC) which outputs an estimate of the message $\hat{m}$. Even though for other schemes soft information (e.g. NewHope) is utilized within the demapping/decoding steps, all procedures have in common that they inherently use the fact that the coefficients of the noise are small in magnitude with high probability.

Definition 3.2. *Let m be the message to be transmitted from the sender to its intended recipient and let $\hat{m}$ be the output of the decoder at the receiver side. We define a decryption failure to be the event that $\hat{m} \neq m$ and denote the probability of this event as the **decryption failure rate (DFR)** of the scheme.*

Remark 3.1. *To reduce the size of the public key, it is common to construct the matrix $\boldsymbol{A}$ with a pseudo-random number generator (PRNG) using a seed obtained from a true random number generator. If the PRNG is cryptographically secure, it is computationally hard to distinguish the resulting matrix $\boldsymbol{A}$ from a uniform sample on $\mathcal{R}_q^{l \times l}$. Within NewHope [AAB+19] and Kyber [ABD+21] SHAKE128 is used as a PRNG to generate $\boldsymbol{A}$.*

3.2.4 Analysis of the Noise

Lemma 3.1. *The distribution of the ciphertext components before compression $(\boldsymbol{u}, v) = (\boldsymbol{A}^T \boldsymbol{s'} + \boldsymbol{e'}, \boldsymbol{s}^T \boldsymbol{A}^T \boldsymbol{s'} + \boldsymbol{e}^T \boldsymbol{s'} + e'' + \mathsf{Encode/Map}(m))$ cannot be distinguished from the uniform distribution on $\mathcal{R}_q^l \times \mathcal{R}_q$ if the decisional MLWE problem is hard for the respective parameter set (n, q, k, l). A similar statement holds for RLWE based schemes if the decisional RLWE problem is hard for the parameter set (n, q, k).*

Proof. The proof of this statement for RLWE/MLWE-based schemes is similar to the security proof in the binary case in [LP11]. We recapitulate it here for the sake of completeness. The result for the RLWE case follows by setting $l = 1$.

Writing the ciphertext tuple before ciphertext compression $\tilde{c}$ as a column vector we obtain

$$\tilde{c} = \begin{pmatrix} \boldsymbol{u} \\ v \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} \\ \boldsymbol{b}^T \end{pmatrix} \boldsymbol{s'} + \begin{pmatrix} \boldsymbol{e'} \\ e'' \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} \\ \mathsf{Encode/Map}(m) \end{pmatrix} .$$

Due to the MLWE assumption $(\boldsymbol{A}, \boldsymbol{b})$ cannot be distinguished from a uniform sample on $\mathcal{R}_q^{l\times l} \times \mathcal{R}_q^l$. Therefore, by Definition 2.45 $\boldsymbol{b}^T\boldsymbol{s'} + e''$ can just be considered to be an additional sample in (2.72). Thus, the ciphertext component v is indistinguishable from a uniformly distributed element in $\mathcal{R}_q$ by the MLWE assumption. Indistinguishability of $\boldsymbol{u}$ holds because $\boldsymbol{u}$ is by definition an MLWE sample. ■

We consider the preimages for the output of the concatenation of compression and decompression function of some input variable z, i.e.

$$\mathcal{Z}_j := \{z \in \mathbb{Z}_q : \mathsf{decomp}_q(\mathsf{comp}_q(z, d_z), d_z) = j\} \ . \tag{3.4}$$

The sets $\mathcal{Z}_j$ partition the set of possible inputs of the compression function $[0, q-1]$ into disjoint sets, which are determined by the target bitlength of the compression d_z.

Corollary 3.1. *Let* $\boldsymbol{u''} = \mathsf{decomp}_q(\mathsf{comp}_q(\boldsymbol{u}, d_u), d_u)$ *and* $v'' = \mathsf{decomp}_q(\mathsf{comp}_q(v, d_v), d_v)$ *be the output of the concatenation of ciphertext compression and decompression for the ciphertext components* $\boldsymbol{u}$ *and* v*, respectively. Then it holds that the problem of distinguishing* $Pr(\boldsymbol{u}|\boldsymbol{u''})$ *and* $Pr(v_i|v_i'')$ *from the uniform distributions on the preimages of* $\mathsf{decomp}_q(\mathsf{comp}_q(\boldsymbol{u}, d_u), d_u)$ *and* $\mathsf{decomp}_q(\mathsf{comp}_q(v_i, d_v), d_v)$*, respectively, can be reduced to the hardness of the decisional MLWE problem for* (n, q, k, l)*.*

Furthermore, let $z'' = \mathsf{decomp}_q(\mathsf{comp}_q(z, d_z), d_z)$ *be the output of the concatenation of ciphertext compression and decompression for an input* $z \in \mathbb{Z}_q$*. Then it holds that* $Pr(z|z'') = 1/|\mathcal{Z}_j|$*.*

Proof. According to Lemma 3.1 all values for v_i can be considered equiprobable. Let the set of preimages of $\mathsf{decomp}_q(\mathsf{comp}_q(v_i, d_v), d_v)$ be denoted by $\mathcal{V}_{i,j}$. Each element $v_i \in \mathcal{V}_{i,j}$ leads per definition to the same output after applying the concatenation of compression and decompression to it. Therefore, it holds for $v_i \in \mathcal{V}_{i,j}$ that $Pr(v_i|v_i'')$ is equal to the uniform distribution on the set $\mathcal{V}_{i,j}$. Virtually the same argument holds for $Pr(\boldsymbol{u}, \boldsymbol{u''})$. ■

Since we have a discrete setting the interval $[0, q-1]$ cannot be subdivided into intervals of equal integer length (q is in fact prime for both NewHope and Kyber), e.g., every v_i is in some unique $\mathcal{V}_{i,j}$ and to each $\mathcal{V}_{i,j}$ there is an interval $\mathcal{A}_j$ associated such that $v_i = j + a$ with $a \in \mathcal{A}_j$.

Recall that the distributions of $\boldsymbol{u}$ and v cannot be distinguished from the uniform distributions on $\mathcal{R}_q^l$ and $\mathcal{R}_q$, respectively, according to Lemma 3.1. Hence, the distributions of the compression noise terms can be computed coefficient-wise. The resulting distribution can be numerically computed by creating a histogram of the compression noise for the procedure of using each $z \in \{0, \ldots, q-1\}$ once at the input of the concatenation compression and decompression and dividing the resulting vector by q.

To compute the distribution of the noise we use the fact that the distribution of the sum of independent variables can be computed by convolving their respective

distributions. Thus, we show in the following that that the terms $\boldsymbol{e}^T\boldsymbol{s'}$, $\boldsymbol{s}^T(\boldsymbol{e'}, \boldsymbol{c_{N_u}})$ and c_{N_v} are stochastically independent with overwhelming probability.

Lemma 3.2. *The set of terms $\boldsymbol{e}^T\boldsymbol{s'}$, $\boldsymbol{s}^T(\boldsymbol{e'}, \boldsymbol{c_{N_u}})$ and c_{N_v} is stochastically independent if there exists a triple of indices $(i, j, w) \in \{1, \ldots, l\}^3$ such that $s_i s'_w \neq 0$ and $s_j s'_w \neq 0$.*

Proof. Recall that $\boldsymbol{u} = \boldsymbol{A}^T\boldsymbol{s'} + \boldsymbol{e'}$. It holds that

$$(\boldsymbol{A}^T\boldsymbol{s'})_i = \sum_{j=1}^{l} A_{ji} s'_j \ . \tag{3.5}$$

If one of the polynomials $s'_j \neq 0$ for $j \in \{1, \ldots, l\}$, the sum in the equation above is uniformly distributed on $\mathcal{R}_q$. This can be shown by considering that we have a uniformly distributed summand for each polynomial coefficient and the assertion follows in accordance with the security proof for a one-time pad. Since the A_{ji} are uniformly distributed, the $\boldsymbol{c_{N_u}}$ is independent of $\boldsymbol{e}$, e'' and $\boldsymbol{s}$ and furthermore independent of $\boldsymbol{s'}$ and $\boldsymbol{e'}$ under the condition that $\boldsymbol{s'} \neq 0$.

In a very similar fashion it follows for c_{N_v} that it is independent of $\boldsymbol{e'}$ and independent of $\boldsymbol{s}, \boldsymbol{s'}, \boldsymbol{e}, e''$ if a pair $(i, j) \in \{1, \ldots, l\}^2$ exists such that $s_i s'_j \neq 0$ by examining

$$\boldsymbol{s}^T\boldsymbol{A}^T\boldsymbol{s'} = \sum_{i=1}^{l}\sum_{j=1}^{l} s_i A_{ji} s'_j \ . \tag{3.6}$$

Since $s_i s'_j \neq 0$ there is a non-zero coefficient within this product, say the coefficient with index p. Notice that this coefficient is multiplied with different coefficients of A_{ji} for each coefficient of the product $s_i A_{ji} s'_j$.

Next we show the independence of the compression noise c_{N_v} with respect to the other noise terms. The existence of indices i, j, w such that $s_i s'_w \neq 0$ and $s_j s'_w \neq 0$ implies that c_{N_v} is decoupled from $\boldsymbol{c_{N_u}}$ by the one-time pad property. This holds because uniformly distributed elements of different rows in $\boldsymbol{A}^T$ contribute to the sum in equation (3.6). ■

Corollary 3.2. *Let an RLWE based scheme with parameters (n, q, k) and only ciphertext compression in v be given. If it holds that $ss' \neq 0$, then the noise distribution can be computed by convolving the distributions of the difference nosie terms and the compression noise c_{N_v}.*

Proof. This statement can be proved similarly to the statement for the compression noise c_{N_v} of Lemma 3.2. ■

The necessary conditions for Lemma 3.2 and Corollary 3.2 hold with overwhelming probability for the parameter sets of Kyber and NewHope considered in this work. Therefore, we neglect the unlikely event that the conditions for Lemma 3.2 or Corollary 3.2 do not hold in the following. The analysis for the validity of this statement can be found in Section 3.A and Section 3.B.

3.2.5 Transforming the Public Key Encryption Scheme into an IND-CCA2 Secure KEM

A common requirement for key-encapsulation mechanisms (KEMs) is IND-CCA2-security. As a reference for explanation of different security notions we refer to [KL21]. In [HHK17] a modular analysis of the Fujisaki-Okamoto transform [FO99] is presented which enables the transformation of an IND-CPA secure PKE scheme into an IND-CCA secure KEM. The authors also address the event of having a non-zero the decryption failure rate. To obtain the desired security level the authors mention that a small decryption failure rate is still very important. For a security level equivalent to AES256 for NewHope1024 [AAB+19] the decryption failure rate is upper bounded by 2^{-216} and for Kyber1024 [ABD+21] the decryption failure rate is upper bounded by 2^{-174}.

3.3 The RLWE/MLWE Channel with Increased Alphabet Size

3.3.1 Channel Model

The encryption and decryption procedure of RLWE-based and MLWE-based cryptosystems can be interpreted as the transmission of symbols over a communication channel with additive noise. The corresponding channel models which we call the RLWE channel and the MLWE channel are illustrated in Fig. 3.1. The sender's goal is to transmit a message m contained in the message space $\mathcal{M}$ reliably to the receiver.

The input to this channel as well as its output is a polynomial in $\mathcal{R}_q$. The additive noise on the channel follows the same distribution as the noise term within equation (3.3). Notice that s, e, s', e' are in $\mathcal{R}_q$ for the RLWE channel while $\boldsymbol{s}, \boldsymbol{e}, \boldsymbol{s'}, \boldsymbol{e'}$ are in $\mathcal{R}_q^l$ for the MLWE channel; e'' is in $\mathcal{R}_q$ in both cases. The crucial properties for the analysis to follow are the same for RLWE and MLWE based systems.

Estimating DFRs for currently proposed schemes (2^{-174} for Kyber, 2^{-216} for NewHope, cf. Section 3.2.5) using Monte Carlo simulations is infeasible. However, it is possible to compute the marginal distribution of the coefficient failure rate for one coefficient $\Pr(\mathcal{E})$. We will show how to obtain an upper bound on $\Pr(\mathcal{E})$ in Section 3.3.3 (Theorem 3.2).

3.3.2 Stochastic Independence Assumption

So far, it is unknown how to precisely obtain the DFR for RLWE/MLWE-based schemes. RLWE/MLWE channels have memory within each message block as the noise is generated from multiplying and adding several polynomials. The coefficients

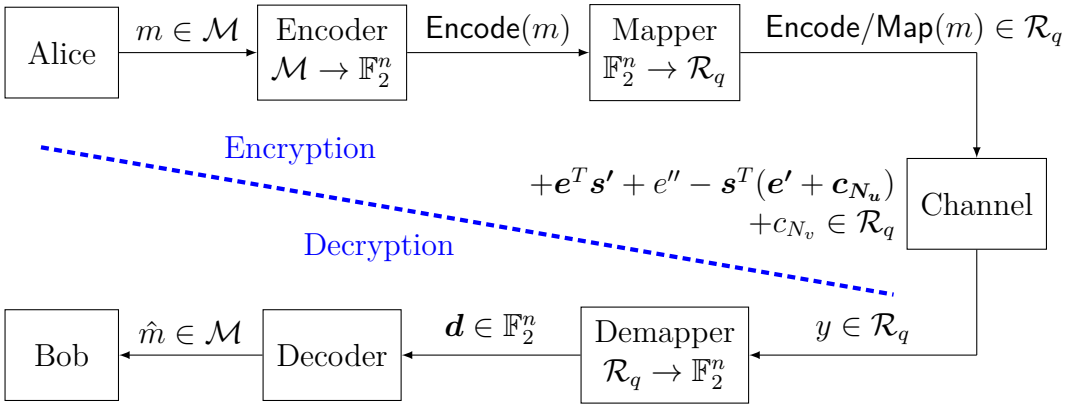

Figure 3.1: RLWE/MLWE channel

of these polynomials are therefore not stochastically independent due to the multiplications, implying that coefficient failures are not independent either. To estimate the DFR, it is widely assumed that coefficient failures within a block occur independently with probability $\Pr(\mathcal{E})$ (cf. [Saa17; FPS18; LLJ+17b]). The assumption of independent coefficient failures is not only common for lattice-based schemes but also for code-based schemes (e.g., HQC [MAB+18], which is an alternative finalist in the NIST-PQC Round 3). Thus the RLWE/MLWE channel can be modelled as n parallel *Binary Symmetric Channels* with error probability $\Pr(\mathcal{E})$. To the best of our knowledge, due to the algebraic operations that connect the different components of the noise in a complicated manner there is no tight bound on the DFR that fully covers the dependency of the coefficient failures. Known upper bounds on the DFR not using this independence assumption are rather loose even though for NewHope the attempt presented in [Sch20] is worth mentioning but not applicable for the encoding schemes presented in this work.

If algebraic codes with hard decision decoding are considered, the minimum distance d determines the number of errors $t = \lfloor (d-1)/2 \rfloor$ up to which correct decoding can be guaranteed. Thus, if we consider stochastically independent coefficient failures an error-correcting code with error-correction capability of t symbols leads to an overall failure rate of the scheme

$$\mathrm{DFR} \leq \sum_{j=t+1}^{n} \binom{n}{j} \Pr(\mathcal{E})^j (1 - \Pr(\mathcal{E}))^{n-j} \ . \tag{3.7}$$

In general however, the the assumption of independent coefficient failures does not hold in RLWE/MLWE-based schemes and it has been shown in [DVV19] that the stochastic dependence between coefficient failures has a significant impact on the DFR of LAC [LLJ+17b]. Hence, the LAC team changed the error distribution for polynomials in their Round 2 submission for the NIST-PQC [LLJ+17a]. In [MFS20] it has

been experimentally shown that this significantly reduces the stochastic dependence of coefficient failures. Quantitative statements have been obtained by using statistical methods.

Although we know that estimating the DFR by using the assumption of independent coefficient failures is not completely accurate we consider it to be a reasonable first order approximation of the real behavior of RLWE/MLWE-based cryptosystems. We state clearly throughout this work whenever we make use of this assumption.

3.3.3 Generalization to Q-ary Alphabets

It is natural to extend the channel input alphabet to be Q-ary. Consider the encoding procedure. In all of the discussed realizations of the RLWE/MLWE schemes, the ECC was binary and the message was mapped to a polynomial with coefficients in $\{0, \lfloor q/2 \rfloor\}$. We extend the channel to Q-ary alphabets by splitting $[-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]$ into smaller intervals of size either $\lfloor q/Q \rfloor$ or $\lceil q/Q \rceil$, where their respective occurrence depends on the remainder of the integer division of q by Q. This approach has already been followed for the LWE-based scheme Frodo in [BCD+16]. Notice that since we are in $\mathbb{Z}_q$ distance is defined to be cyclic. Therefore, it is perfectly fine that $-\lfloor q/2 \rfloor$ and $\lfloor q/2 \rfloor$ belong to the same quantization interval as they only have a distance of 1 for odd q. The appropriate distance measure for these kinds of considerations is the Lee-metric [Lee58]. It is defined as $d_L(x, y) := \min(|x - y|, q - |x - y|)$.

The channel's input alphabet consists of the central elements of these intervals. The deployed ECC is changed to be of Q-ary alphabet size and the mapper's output alphabet is defined to be equal to the channel's input alphabet.

A possible choice for the demapper is to extend the hard decision demapping procedure of LAC to Q-ary alphabet sizes. The demapping procedure within LAC can be interpreted as a linear quantization of $[-\lfloor q/2 \rfloor, \lfloor q/2 \rfloor]$. Recall the subintervals considered in the construction of the mapper. We define these subintervals to be the quantization intervals and their center points to be the respective reproduction values. By this methodology we have generalized Encode/Map and Demap/Decode for $Q = 2$ to arbitrary Q. Basically, the receiver uses the quantizer to estimate the symbols transmitted by the sender and uses the ECC to correct possibly erroneous symbols. We remark that the choice of the demapper is by no means optimal because soft information is not utilized. Notice that the difference of the decryption function for different alphabet sizes lies entirely in Demap/Decode and in particular Equation (3.3) does not change if we consider the Q-ary case because all the information about the input alphabet size is contained in the functions Encode/Map and Demap/Decode.

The following theorem is based on a result in [LP11] and proves that generalizing RLWE/MLWE-based schemes to $Q \geq 2$ does not necessarily decrease their security level.

Theorem 3.1. *The security level of RLWE/MLWE-based schemes is not reduced by*

the generalization to a Q-ary alphabet as long as the decryption failure rate is not increased.

Proof. Due to Lemma 3.1 it holds that the tuple $(\boldsymbol{u}, v)$ cannot be distinguished from a uniformly distributed sample on $\mathcal{R}_q^l \times \mathcal{R}_q$ under the MLWE assumption. By following exactly the same steps as in its proof, it follows that the value of $\mathsf{Encode}/\mathsf{Map}(m)$ has no influence on the distribution of $(\boldsymbol{u}, v)$ irrespective of Q. As already mentioned in Section 3.2.5 a low decryption failure rate is essential to obtain a high security level for the resulting scheme after the transformation into an IND-CCA secure KEM. Hence, we have to avoid increasing the DFR in order to keep the same security level. ■

Generalizing the RLWE/MLWE-based scheme to Q-ary input alphabets increases the coefficient failure probability if all other parameters of the system remain the same. To avoid increasing the DFR, the error-correction capability of the deployed ECC has to be increased accordingly.

We consider the generalization of the demapping strategy of LAC for the Q-ary case and we show how to upper bound the coefficient failure probability $\Pr(\mathcal{E})$ which can then be used to obtain an upper bound on the decryption failure rate similar to (3.7). We define ψ to be the probability distribution of the i-th coefficient of the noise $(\boldsymbol{e}^T\boldsymbol{s'} - \boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c_{N_u}}) + e'' + c_{N_v})_i$. Indexing the distribution ψ is unnecessary in both cases because all coefficients of the difference noise are distributed in the same way due to the symmetry of χ_k.

In order to compute an upper bound on $\Pr(\mathcal{E})$ for RLWE/MLWE channels we first prove the following Lemma.

Lemma 3.3 (Noise distribution for MLWE)**.** *Recall that the noise is given by* $\boldsymbol{e}^T\boldsymbol{s'} + e'' - \boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c_{N_u}}) + c_{N_v}$*. We define the distribution of the product of two elements in* $\mathbb{Z}_q$ *which have been sampled according to the error distribution* χ_k *by* ξ_k*. Furthermore, we define the distribution of one coefficient of* $\boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c_{N_u}})$ *by* η_k *and the distribution of* c_{N_v} *by* ρ_v*. Then for the MLWE channel it holds*

$$\psi = \circledast_{l-1}(\circledast_{n-1}(\xi_k)) * \eta_k * \chi_k * \rho_v \tag{3.8}$$

Proof. Consider the product of two polynomials $a, b \in \mathcal{R}_q$ sampled according to $\chi_k(\mathcal{R}_q)$. The i-th coefficient of their product equals

$$(ab)_i = \sum_{j=0}^{i} a_j b_{i-j} - \sum_{j=i+1}^{n-1} a_j b_{n-j+i} \ . \tag{3.9}$$

We remark that addition and subtraction of polynomials sampled according to ξ_k or χ_k leads to the same resulting distributions due to the symmetry of the distribution χ_k around zero. Since the first summand of the noise is $\boldsymbol{e}^T\boldsymbol{s'}$ we obtain its distribution by first summing n terms that are distributed according to ξ_k for one polynomial

multiplication and then summing l terms that are distributed according to the resulting distribution to compute the scalar product. To obtain the overall noise the result is added to coefficient of $\boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c_{N_u}})$ which is distributed according to η_k, a coefficient of e'' which is distributed according to χ_k and a coefficient of c_{N_v} distributed according to ρ_v. The assertion follows from these considerations. Notice that the computation of the noise distribution makes use of Lemma 3.2. ■

The distribution η_k of a single coefficient of $\boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c_{N_u}})$ can be easily computed for the parameter set of Kyber that we consider in this work (Kyber1024).

Remark 3.2. *For NewHope it holds that* $\psi = \circledast_{n-1}(\xi_k) * \circledast_{n-1}(\xi_k) * \chi_k * \rho_v$ *because the first component of the ciphertext is not compressed, i.e.* $c_{N_u} = 0$.

Theorem 3.2. *Let the alphabet size be* Q *and let the probability distribution of a coefficient of the difference noise be denoted again by* ψ. *Then the length of every demapping (quantization) interval is at least* $\lfloor q/Q \rfloor$ *and it holds that*

$$\Pr(\mathcal{E}) \leq 1 - \sum_{i=-\lfloor q/(2Q) \rfloor}^{\lfloor q/(2Q) \rfloor} \psi(i) =: \overline{\Pr(\mathcal{E})} \ . \tag{3.10}$$

Assuming coefficient failures to occur stochastically independent with respect to each other it follows that

$$DFR \leq \sum_{j=t+1}^{n} \binom{n}{j} \overline{\Pr(\mathcal{E})}^j (1 - \overline{\Pr(\mathcal{E})})^{n-j} \ . \tag{3.11}$$

Proof. We will show this statement by proving that the probability of a successful reception is lower bounded by

$$\sum_{i=-\lfloor q/(2Q) \rfloor}^{\lfloor q/(2Q) \rfloor} \psi(i) \ . \tag{3.12}$$

Indeed, if we choose the quantization intervals according to section 3.3.3 and put the reconstruction values into the middle of the intervals we obtain that the probability for a correct symbol is lower bounded by (3.12). This statement implies the upper bound given in (3.10). Inequality (3.11) follows from (3.10) by using a standard combinatorial argument. ■

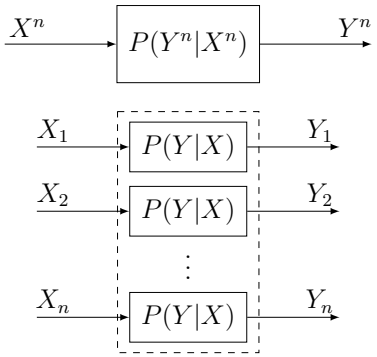

Figure 3.2: RLWE/MLWE and marginalized RLWE/MLWE channel (dashed)

3.4 Information-Theoretic Analysis of the RLWE/MLWE Channel

3.4.1 A Lower Bound on the Capacity of the RLWE/MLWE-Channel

We define an auxiliary channel consisting of n parallel channels which are defined by the marginalization of $P(Y^n|X^n)$ for one coefficient. We label its distribution by $P(Y|X)$, where the distribution is independent of the index within the RLWE/MLWE block due to the symmetry of χ_k. This auxiliary channel is in the following referred to as the *marginalized RLWE/MLWE channel*. Fig. 3.2 depicts RLWE/MLWE channel and marginalized RLWE/MLWE channel. The channel inputs $X_1 \dots X_n$ denote the coefficients of the polynomial $\mathsf{Encode}/\mathsf{Map}(m)$ and the corresponding channel outputs $Y_1 \dots Y_n$ denote the coefficients of $y \in \mathcal{R}_q$.

Lemma 3.4. *Let X_i denote the i-th input symbol to the RLWE/MLWE channel and let Y_i denote the i-th output symbol of the channel. We denote the vectors containing the sequences $X_1, \dots, X_k$ and $Y_1, \dots, Y_k$ by X^k and Y^k, respectively. Let $H(X_i) = H(X_j)$ and let $H(X_i|Y_i) = H(X_j|Y_j)$ $\forall i, j$. Furthermore, let the input symbols to the channel be stochastically independent, then*

$$I(X^n; Y^n) \geq nI(X; Y) \ , \tag{3.13}$$

where we omitted the indices on the right hand side of the previous inequality because the mutual information between $I(X_i; Y_i)$ does not depend on the respective index i.

Proof.

$$\begin{aligned} I(X^n;Y^n) &= H(X^n) - H(X^n|Y^n) \\ &= nH(X) - \sum_{i=1}^{n} H(X_i|Y^n X^{k-1}) \\ &\geq nH(X) - \sum_{i=1}^{n} H(X_i|Y_i) \\ &= nH(X) - nH(X|Y) = nI(X;Y) \end{aligned} \tag{3.14}$$

■

Since the marginalized RLWE/MLWE channel is composed of n identical component channels its capacity can be computed by determining the capacity of one component channel and multiplying the result by n. Let X be a random variable modelling the input distribution of one component channel and Y be the random variable specifying its output. We denote the ranges of X and Y by $\mathcal{X}$ and $\mathcal{Y}$, respectively, where $|\mathcal{X}| = Q$.

Lemma 3.5. *Each component channel of the marginalized RLWE/MLWE channel belongs to the class of uniformly dispersive channels, meaning that the set* $\{P(y|x) : y \in \mathcal{Y}\}$ *is the same for all* $x \in \mathcal{X}$ *and it holds that*

$$H(Y|X) = H(Y|X = x) = H(\psi) \tag{3.15}$$

for all $x \in supp(P_X)$ *where* ψ *denotes the distribution of one coefficient according to the channel noise (Lemma 3.3).*

Proof. Recapping the definition of the conditional entropy (Equation (2.10)) we have

$$H(Y|X) = \sum_{x \in \mathrm{supp}(P_X)} H(Y|X = x) P_X(x) \tag{3.16}$$

Without loss of generality we assume that we analyze the i-th component channel and therefore its output $Y_i = (\mathsf{Encode/Map}(m))_i + (\boldsymbol{e}^T\boldsymbol{s'} - \boldsymbol{s}^T(\boldsymbol{e'} + \boldsymbol{c}_{\boldsymbol{N_u}}) + e'' + c_{N_v})_i$, where $(\mathsf{Encode/Map}(m))_i$ denotes the i-th component of the encoded message after the mapper.

Since $X_i = (\mathsf{Encode/Map}(m))_i$ it follows that $P_{Y|X=x} = \psi,\ \forall x \in \mathrm{supp}(P_X)$ and therefore $\{P(y|x) : y \in \mathcal{Y}\}$ is the same $\forall x \in \mathrm{supp}(P_X)$ which implies
$H(Y|X = x) = H(\psi),\ \forall x \in \mathrm{supp}(P_X)$. Applying this result to (3.16) proves the first equality of this lemma. ■

The following corollary is standard textbook knowledge in the field of information theory and can for instance be found in [Mas98]. We recap it here for the the sake of completeness.

Lemma 3.6. *For uniformly dispersive channels the channel capacity is equal to*

$$C = \max_{P_X} H(Y) - H(Y|X = a) \tag{3.17}$$

for some $a \in \mathrm{supp}(P_X)$.

Lemma 3.6 states that for uniformly dispersive channels the maximization of the mutual information boils down to the maximization of an entropy. Therefore, we aim at finding P_X such that $H(Y)$ is maximized. We start by stating the following lemma which is given as an exercise in [CT99].

Lemma 3.7. *Consider a random variable* Y *with distribution* P_Y *and consider a random variable* Z *with the same distribution except for* k *events, denoted as* $a_{i_1}, \ldots, a_{i_k}$*, where* $P_Z(a_{i_1}) = \cdots = P_Z(a_{i_k}) = \sum_{j=1}^{k} P_Y(a_{i_j})/k$*. Then it holds that* $H(Y) \leq H(Z)$.

Next we make use of Lemma 3.7 to show that the uniform distribution achieves capacity if Q divides q.

Theorem 3.3. *If* q *is divisible by* Q *the uniform distribution on* $\mathcal{X}$ *achieves the capacity of a component channel of the marginalized RLWE/MLWE channel.*

The distribution of the output Y *can then be computed by:*

$$P_Y(y) = \frac{1}{Q} \sum_{j=0}^{Q-1} \psi\left(y + \frac{jq}{Q}\right) \tag{3.18}$$

Proof. We know due to Lemma 3.6 that the problem of maximizing $I(X;Y)$ can be reduced to maximizing the entropy of the output distribution $H(Y)$. Since any finite dimensional cube $[0,1]^Q$ is compact and the distribution $H(Y)$ is a continuous function with respect to P_X we know that there exists some P_X^* maximizing $H(Y)$. Suppose that we are provided with this distribution. We take the resulting output distribution P_Y^* and observe its values for the set $\{i, i + q/Q, i + 2q/Q, \ldots, i + (Q-1)q/Q\}$.

$$\begin{aligned}
P_Y^*(i) &= P_X^*(0)\psi(i) + P_X^*(1)\psi\left(i - \tfrac{q}{Q}\right) + \ldots \\
&\quad + P_X^*(Q-1)\psi\left(i - \tfrac{(Q-1)q}{Q}\right) \\
P_Y^*\left(i + \tfrac{q}{Q}\right) &= P_X^*(0)\psi\left(i + \tfrac{q}{Q}\right) + P_X^*(1)\psi(i) + \ldots \\
&\quad + P_X^*\left(Q-1\right)\psi\left(k - \tfrac{(Q-2)q}{Q}\right) \\
&\vdots \\
P_Y^*\left(i + \tfrac{(Q-1)q}{Q}\right) &= P_X^*(0)\psi\left(i + \tfrac{(Q-1)q}{Q}\right) + \ldots \\
&\quad + P_X^*(Q-1)\psi(i)
\end{aligned}$$

Notice that the set of arguments within the function ψ is the same for each equation. Furthermore, each possible pair of arguments with respect to P_X^* and P_Y^* occurs exactly once if all above equations are considered. Lemma 3.7 states that for some specific i we can only increase the entropy $H(Y)$ if we change P_Y^* by replacing all values $P_Y^*(i), \ldots, P_Y^*\left(i + (Q-1)q/Q\right)$ with their average without changing the remaining values of P_Y^*. This procedure can be performed for every i without decreasing $H(Y)$ and is equivalent to changing P_X^* to be the uniform distribution on the set $\mathcal{X}$ which concludes the proof. ■

Using Lemma 3.4 we give a lower bound on the channel capacity of the RLWE/MLWE channel.

Theorem 3.4. *Let n denote the RLWE/MLWE blocklength, let X_i and Y_i be the random variables specifying the i-th channel input and output, respectively, and let $P_X^{\otimes n}$ be the set of the product distributions on X^n, meaning that $X_1, \ldots, X_n$ are independent identically distributed random variables according to some probability distribution P_X. Recall that $\psi = \circledast_{l-1}(\circledast_{n-1}(\xi_k)) * \eta_k * \chi_k * \rho_v$ denotes the distribution of one coefficient of the noise for the RLWE/MLWE channel (Lemma 3.3).*

The capacity of the RLWE/MLWE channel $C_{RLWE/MLWE}$ is lower-bounded by

$$\begin{aligned} C_{RLWE/MLWE} &= \max_{P_{X^n}} I(X^n; Y^n) \geq \max_{P_X^{\otimes n}} I(X^n; Y^n) \\ &\geq \max_{P_X^{\otimes n}} \sum_{i=1}^{n} I(X_i; Y_i) = n \max_{P_X} I(X; Y) =: C_{RLWE/MLWE}^{indep} \ . \end{aligned} \tag{3.19}$$

Furthermore, it holds that

$$C_{RLWE/MLWE}^{indep} \geq n \left(H\left(\frac{1}{Q} \sum_{j=0}^{Q-1} \psi_j \right) - H(\psi) \right) \ , \tag{3.20}$$

where we define $\psi_j(x) := \psi\left(x + \lfloor jq/Q \rfloor\right)$.

Proof. The channel coding theorem states that the capacity of a DMC with input U and output V is equal to $C = \max_{P_U} I(U; V)$. In the context of the RLWE-channel $U \widehat{=} X^n$ and $V \widehat{=} Y^n$ and therefore the first equality in (3.19) follows. The subsequent inequality follows due to the restriction of the maximization from the set of all distributions on X^n to the set of product distributions. Furthermore, as a consequence the requirements for applying Lemma 3.4 are fulfilled because the inputs are independently and identically distributed and $H(X_i|Y_i) = H(X_j|Y_j)\ \forall i, j$ due to the symmetry of the error distribution. This shows the second inequality in (3.19).

Recall that by Lemma 3.6 the last maximization in (3.19) reduces to a maximization of $H(Y)$. Choosing a particular distribution at the input, e.g. the uniform distribution on the set $\mathcal{X}$, leads to a lower bound the channel capacity. For this input distribution

it holds that

$$I(X;Y) = H(Y) - H(Y|X) = H\left(\frac{1}{Q}\sum_{j=0}^{Q-1}\psi_j\right) - H(\psi) \ ,$$

where the entropy of the output distribution can be computed in accordance with the proof of Theorem 3.3 and the conditional entropy is given by Lemma 3.5. ■

We emphasize that Theorem 3.4 shows that the capacity of the RLWE/MLWE channel is lower bounded by the capacity of the marginalized RLWE/MLWE channel which is equal to $\max_{P_X^{\otimes n}} \sum_{i=1}^{n} I(X_i;Y_i)$. This is no contradiction to the results in [DVV19] in which it has been shown that assuming independent coefficient failures leads to an overestimation of the security level for two reasons. The first one is that the decoder used in [DVV19] does not take the channel's memory into account and therefore we are in a mismatched decoding scenario. The second reason why the results are not necessarily comparable with each other is that Theorem 3.4 gives a statement in the asymptotic setting while the results in [DVV19] deal with error correction over a single RLWE block for LAC.

Remark 3.3. *The difference between the optimized mutual information* $\max_{P_X} I(X;Y)$ *and* $I(X;Y)$ *for* P_X *being chosen to be uniform has been experimentally observed not to be significant for the parameter sets considered in this work.*

3.4.2 The Quantized RLWE/MLWE Channel

The demapper implemented in LAC outputs each symbol by making a hard decision. By generalizing this demapper to the Q-ary case as described in Section 3.3.3, it is possible to define a new channel which we refer to as the *quantized RLWE/MLWE channel*. This channel encapsulates the mapper, the RLWE/MLWE channel and the demapper into one channel which can be analyzed similarly to the RLWE/MLWE channel.

In contrast to the RLWE/MLWE channel, soft information can only be used during the decoding process rather than throughout the combined process of demapping and decoding. Therefore, the capacity of the quantized RLWE/MLWE channel is lower than the capacity of the RLWE/MLWE channel. Obtaining a lower bound on the capacity of the quantized RLWE/MLWE channel is very similar to the statements in Theorem 3.4. The fixed demapper just quantizes the output of the RLWE/MLWE channel Y which is distributed according to the probabilities of the quantization intervals. Notice that this channel is not uniformly dispersive if Q does not divide q but the lower bound in (3.19) is still valid.

3.4.3 Plaintext Bits per Ciphertext Bit

Up to now we have shown how to obtain lower bounds on the capacities for the RLWE/MLWE channels and their quantized counterparts. In practical terms it may however be more important how many plaintextbits can be transmitted over the channel per ciphertextbit. A scheme corresponding to a channel with high capacity does not necessarily perform well in terms of plaintextbits per ciphertextbit. This effect occurs because the parameter l as well as the deployed ciphertext compression play an important role for this figure of merit.

Lower bounds on the maximal amount of plaintextbits per ciphertextbit can be computed for both NewHope and Kyber from the lower bounds on the capacities determined by the methodology described in Theorem 3.4.

Proposition 3.1. *Let an MLWE based encryption scheme with ciphertext compression parameters d_u and d_v be given. Then the amount of plaintextbits per ciphertextbit for a given bitrate R is given by the following formula:*

$$\frac{R}{ld_u + d_v}$$

Proof. The amount of plaintext bits for a ciphertext blocklength of n symbols is equal to $R{\cdot}n$. By the definition of the ciphertext compression we have that the corresponding ciphertext is of size $n(l \cdot d_u + d_v)$ bits. Dividing the amount of plaintextbits by the size of the corresponding ciphertext we obtain the statement of the proposition. ■

3.5 Maximizing Asymptotic Rates for NewHope and Kyber

This section shows the examines the influence of Q on NewHope with $(n, q, k, d_u, d_v) = (1024, 12289, 8, 0, 3)$ and Kyber with $(n, q, k, l, d_u, d_v) = (256, 3329, 2, 4, 11, 5)$.

In Fig. 3.3 the lower bounds on the capacity obtained by applying Theorem 3.4 to the aforementioned parameter sets for NewHope and Kyber are plotted for the RLWE/MLWE channel and the quantized RLWE/MLWE channel. For all parameter sets the results show that the lower bounds on the respective channel capacities can be significantly increased if larger input alphabet sizes Q are considered compared to the originally proposed binary case. The plots also show that Q only influences the lower bounds on the capacities for small alphabet sizes. The reason for this is that already for moderate alphabet sizes Q, $P_Y(y)$ is almost uniform such that increasing Q further cannot significantly increase $H(Y)$ and therefore the capacity of the RLWE/MLWE channels since it is uniformly dispersive and therefore $H(Y|X)$ does not depend on Q. In the quantized case a similar effect occurs even though the alphabet size needs to be increased a bit in order to reduce the penalty on the achievable rate due to

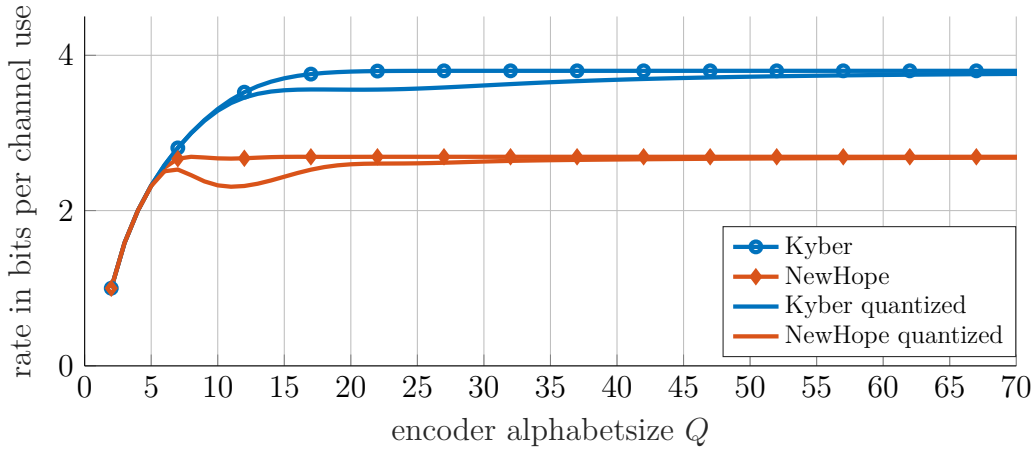

Figure 3.3: Lower bounds on the capacities of the RLWE/MLWE channels according to NewHope and Kyber

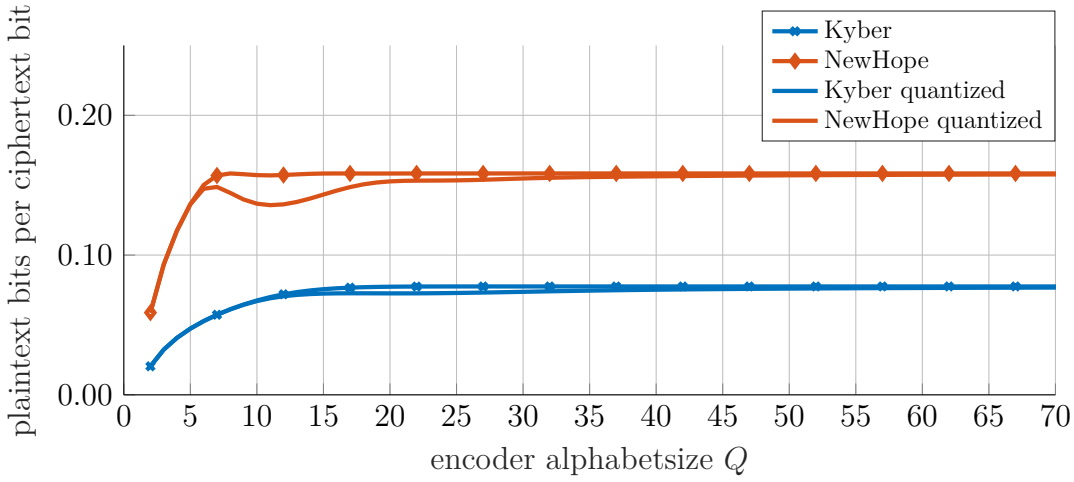

Figure 3.4: Lower bounds on the maximal amount of plaintextbits per ciphertextbits for the RLWE/MLWE channels of NewHope and Kyber

fixing the demapping strategy. In fact the intervals of the demapper are shrinking as Q is increased until eventually each interval only contains one element and quantized RLWE/MLWE channel and RLWE/MLWE channel coincide.

Applying Proposition 3.1 to the lower bounds on the capacity of the RLWE/MLWE channel and the quantized RLWE/MLWE channel, leads to lower bounds on the maximal amount of plaintextbits per ciphertextbit for the respective channels. The resulting bounds are presented in Fig. 3.4 and show that the lower bounds on the maximal amount of plaintextbits per ciphertextbit for NewHope are higher than for Kyber.

3.6 Minimizing Ciphertext and Key Sizes in the Asymptotic Setting

The asymptotic achievability bound (Theorem 3.4) can be used to find parameters that significantly reduce ciphertext and key sizes in the asymptotic regime when a large amount of ciphertext blocks are concatenated and the messages within all blocks are encoded using an error-correcting code (ECC) of suitable blocklength.

In the following, we show how to modify the parameters of Kyber without compromising the schemes' security while still keeping the bitrate to be at least 1. We do this because this is equivalent to encrypting 256 bit within one MLWE block for Kyber. We decided not to modify the parameters n and q because only specific combinations permit the use of the number theoretic transform (NTT) for the required polynomial multiplications. Similarly to the FFT this reduces the multiplication cost of convolutions from $\mathcal{O}(n^2)$ to approximately $\mathcal{O}(n \log n)$.

3.6.1 Decreasing Keys and Ciphertexts by Modifying l

A natural way to decrease both ciphertext and key sizes is to modify the MLWE parameter l. For RLWE based cryptosystems this methodology does not work in a straightforward way because it holds that $l = 1$ for those schemes, i.e. l cannot be modified. However, not changing other parameters accordingly will decrease the security level of the scheme. Our approach to this problem is to increase the parameter k, which specifies the centered binomial distribution, to compensate for this effect. This, in turn, increases the Decryption Failure Rate (DFR), which needs to be compensated by choosing an appropriate ECC. Using the results presented in [APS15], it is possible to estimate the hardness of the underlying lattice problems of the algorithm for the modified parameters. The authors also provide a sage tool that implements their work, which we used to estimate the security of our modified parameters. The lower bound on the capacity of the resulting MLWE channel gives insight into whether such an ECC exists.

In this section, we try to reach the security level of the parameter set Kyber1024 $(n, q, l, d_u, d_v, k_1, k_2) = (256, 3329, 4, 11, 5, 2, 2)$ after the modification of $l = 3$. To compensate for the loss in security, the parameters k_1, k_2 of the centered binomial distribution have to be increased accordingly. We obtained that $k_1 = k_2 = 14$ is the smallest value that achieves the desired level of security.

Fig. 3.5 shows that it is possible to achieve bitrates greater than 1 for this change in parameters if the alphabet size of the encoder is increased appropriately. This parameter set decreases the size of the private key and the ciphertexts by 25%, whereas the number of elements in $\boldsymbol{A}$ is decreased by about 44%. Even though the matrix $\boldsymbol{A}$ is generated from a pseudorandom number generator using a seed of fixed length, the memory consumption on the device running the algorithm is significantly reduced.

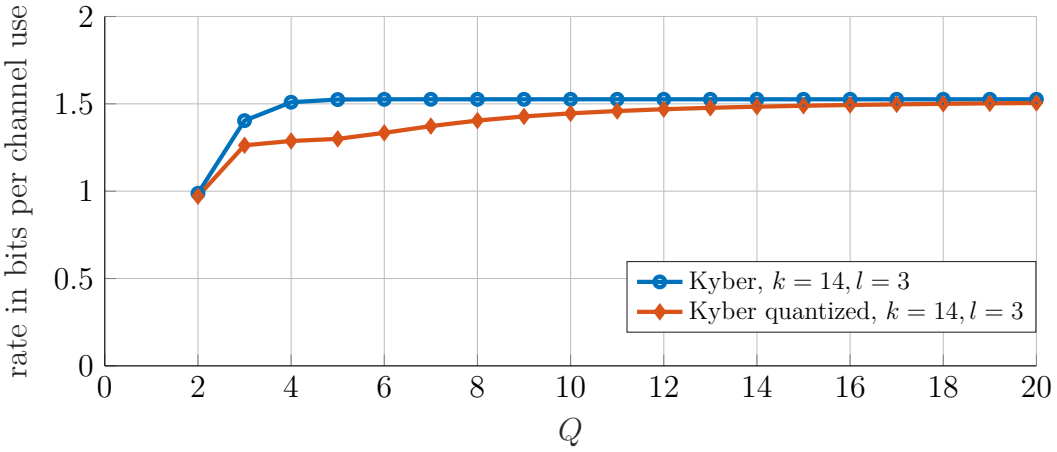

Figure 3.5: Lower bounds on the capacities of the MLWE channel for Kyber with $l = 3$ and $k = 14$ achieving the security of the Kyber1024 parameter set

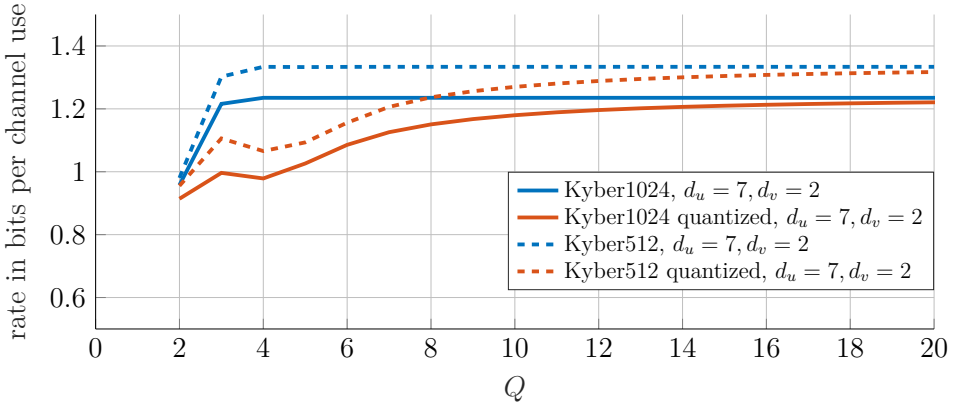

Figure 3.6: Lower bounds on the capacities of the MLWE channels for enhanced ciphertext compression for the Kyber512 and Kyber1024 parameter

3.6.2 Decreasing the Ciphertext Size by Modifying the Compression Parameters d_u and d_v

In the following, we will examine the parameter sets Kyber512 $(n, q, l, d_u, d_v, k_1, k_2) = (256, 3329, 2, 10, 4, 2, 3)$ and Kyber1024 that achieve the smallest and highest security levels proposed within the NIST PQC standardization. The results in Fig. 3.6 show that asymptotically, it is possible to significantly enhance the ciphertext compression by using $d_u = 7$ and $d_v = 2$ for both parameter sets. This reduces the ciphertext by about 39% for Kyber1024 and by about 33% for Kyber512. The difference between the two curves in the figure is that Kyber quantized refers to the case where mapper and demapper are included in the channel. In this case, soft information can only be used during decoding rather than during the combined process of demapping and decoding.

3.7 Achievable Data Rates and DFRs for a Single Ciphertext Block

In Sections 3.5 and 3.6 asymptotic results as the number LWE/RLWE/MLWE blocks goes to infinity are provided. In this case the utilized ECC may be taken over all messages of all blocks combined. In this section we are interested in the amount of plaintextbits that we can transmit using a single ciphertext block such that the desired DFR of the respective scheme is achieved.

3.7.1 Maximizing the Achievable Rate using BCH Codes

Our goal in this section is to maximize the achievable rate for a single RLWE/MLWE block by varying Q using error correcting codes to achieve the required DFRs for NewHope and Kyber under the assumption of stochastically independent coefficient failures. We use Theorem 3.2 to find the required minimum distances d for Q-ary codes that guarantee the required decryption failure rates given in the supporting documentations of NewHope (less than 2^{-216}) [AAB+19] and Kyber (less than 2^{-174}) [ABD+21]. We use the Gilbert–Varshamov bound to show that there exists a Q-ary linear code with minimum distance d and dimension k_{GV} for a given RLWE/MLWE blocklength n. We compute the resulting bitrate according to

$$R_{GV} := \frac{k_{GV}}{n} \log_2(Q) \ . \tag{3.21}$$

Notice, however, that the Gilbert–Varshamov bounds only states that codes with parameters $[n, k_{GV}, d]_Q$ exist. In a second step, we therefore examine BCH codes of length at most n achieving at least the required minimum distance d. These codes can be explicitly constructed and efficiently encoded and decoded. There are constraints on the length of BCH codes depending on their field size, e.g., there are no even length binary BCH codes. Since we cannot reduce the length of one RLWE block n as this would have a negative effect on the security level of the scheme, we define the rates of BCH codes with respect to n, i.e.

$$R_{BCH} := \frac{k_{BCH}}{n} \log_2(Q) \ . \tag{3.22}$$

For the parameter set $(n, q, k) = (1024, 12289, 8)$ of NewHope, Table 3.1 shows that $Q = 4$ is optimal and that there is a BCH code achieving a bitrate $R_{BCH} = 1.7813$ at a DFR $< 2^{-216}$. Table 3.2 shows that for Kyber the optimal alphabet size is $Q = 5$ and the best BCH code achieves a bitrate of 1.8412 for DFR $< 2^{-174}$. For NewHope the original proposal achieves a rate of 0.25 whereas for Kyber the original proposal does not deploy any ECC and therefore its bitrate equals 1. The results for the best binary BCH code are omitted because an ECC is not necessary to achieve the required

Table 3.1: Code parameters for different Q for NewHope with DFR $< 2^{-216}$, $(n, q, k) = (1024, 12289, 8)$

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
2	3	1014	0.9902	1023	1013	0.9893	0.0582
3	11	973	1.5060	1022	949	1.4689	0.0864
4	31	907	1.7715	1023	912	1.7813	0.1048
5	81	784	1.7777	939	554	1.2562	0.0739
7	369	344	0.9431	960	91	0.2495	0.0147

Table 3.2: Code parameters for different Q for Kyber with DFR $< 2^{-174}$, $(n, q, k, l) = (256, 3329, 2, 4)$

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
2	1	256	1	–	–	–	0.0204
3	5	240	1.4859	242	231	1.4302	0.0292
4	9	228	1.7813	255	231	1.8047	0.0368
5	15	214	1.9410	252	203	1.8412	0.0376
7	33	180	1.9739	240	143	1.5682	0.0320

DFR. However, for larger alphabet sizes higher rates can be achieved by using BCH codes. The results for all aforementioned schemes show that a substantial increase in bit rate is achievable by increasing Q and using suitable ECCs.

The capacity of the quantized RLWE/MLWE channel is an upper bound on the achievable rate by the concrete BCH code constructions. For NewHope and $Q = 4$ there is a gap of about 0.2 bit per channel use between the lower bound on the capacity and the rate achieved by the BCH code. Similarly this gap is about 0.5 bit per channel use for Kyber for $Q = 5$.

3.7.2 Minimizing the DFR for a Given Minimum Rate

Designing NewHope in a way that the bit rate equals 0.25 makes sense because it enables the transmission of 256 bit of information within one RLWE block with $n = 1024$. This corresponds to one AES256 key [DR98]. Similarly for Kyber the achieved rate has to be at least 1 because for this scheme $n = 256$. Public key algorithms are often used to share the key of a symmetric cryptosystem because those can be implemented very efficiently in hardware and symmetric algorithms are usually able to perform encryption quicker and without any ciphertext expansion. Due to Grover's

Table 3.3: Decryption failure rates for different alphabet sizes Q for a BCH-bitrate of at least 0.25 for NewHope and 1 for Kyber

Q	$d_{NewHope}$	d_{Kyber}	NewHope	Kyber
2	214	1	2^{-12769}	2^{-174}
3	213	26	2^{-4307}	2^{-989}
4	424	46	2^{-3646}	2^{-953}
5	299	44	2^{-1075}	2^{-547}
7	366	59	2^{-213}	2^{-338}

algorithm [Gro96] the brute-force search of the key can be done in $\mathcal{O}(2^{N/2})$ where N denotes the length of the key in bit. Therefore, in order to obtain a post-quantum security level of 128 bit a key having a length of at least $N = 256$ bit is required for AES.

However, if the the public key encryption (PKE) scheme shall directly be used to encrypt data or if in the future another symmetric cryptosystem with longer key size is used, it is sensible to transmit more data per ciphertext block. This can be useful if one would like to avoid an extra AES implementation to save chip area. Additionally, it is possible to share longer symmetric keys if that is necessary in the future using the same PKE system parameters.

Table 3.3 shows the largest minimum distances for NewHope and Kyber that achieve the required BCH-bitrates of 0.25 and 1, respectively, for different alphabet sizes Q. Furthermore the resulting DFRs of the schemes under the assumption of independent coefficient failures are given. We observe that for NewHope $Q = 2$ gives the lowest DFR $= 2^{-12769}$ whereas for Kyber $Q = 3$ is optimal resulting in DFR $= 2^{-989}$. Notice that the optimal alphabet sizes for minimizing the DFR are different from the optimal alphabet sizes for maximizing the achievable rates for the required DFRs in NewHope and Kyber.

3.7.3 Coding over Multiple Ciphertext Blocks

In order to achieve capacity, it is in general necessary to perform coding over infinitely long blocks. This is of course not possible but it was shown in [Ari09] that polar codes are capacity-achieving for the class of binary memoryless symmetric (BMS) channels. Furthermore, it has been shown in [KKM+17] that extended primitive narrow-sense BCH codes are capacity-achieving on the binary erasure channel (BEC) for blockwise MAP decoding. Even though the channel that we analyze in this work is no BEC we are still using BCH codes to obtain results in the finite length regime. The reason for this is that the claimed DFRs within the schemes are essential for the claimed security levels. Capacity-achieving codes using iterative decoding approaches like LDPC or polar codes

Table 3.4: Code parameters for different Q for Kyber with DFR $< 2^{-174}$, $(n, q, s, l) = (256, 3329, 2, 4)$ for coding over 4 blocks

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
3	5	1004	1.5540	1022	997	1.5432	0.0315
4	9	989	1.9316	1023	993	1.9395	0.0396
5	17	963	2.1836	1008	899	2.0385	0.0416
7	41	904	2.4784	960	757	2.0754	0.0424
8	59	866	2.5371	1023	738	2.1621	0.0441
9	87	811	2.5106	1022	631	1.9533	0.0399

cannot be simulated down to the DFRs we require for the systems under consideration in this work (e.g. for Kyber 2^{-174}). In particular, LDPC codes feature error floors and thus need to be simulated down to the required DFR which is not feasible for the required DFRs of Kyber or NewHope. For polar codes there exist bounds on the DFR [RTA18], [STA09] and therefore those are more suitable for LWE/RLWE/MLWE based cryptosystems. Using those upper bounds on the DFR reduces the achievable rate though and the DFRs we are aiming at are very small. We do not investigate polar codes further throughout this work but this could be an interesting point for further research if the bounds are tight enough.

It is possible to use BCH codes over more than one ciphertext block to increase the achievable rate bringing it closer to the channel's capacity. In this work, we took the approach to perform coding over four RLWE/MLWE blocks. This choice is arbitrary and has no particular reason, rather we took it as an example. The results of this approach are presented within Tables 3.4 and 3.5. Compared to the results in Section 3.7.1 we observe that coding over multiple blocks increases the achievable rates significantly, especially for larger input alphabet sizes. Notably, for Kyber the highest achievable rate is increased from 1.7813 for $Q = 4$ to 2.0754 for $Q = 7$. Notice also that for all Q the achievable rate is increased and thereby closer to the lower bound on the channel capacity shown in Fig. 3.3. For the other presented schemes similar behavior is observed.

3.8 Finite-Length Achievability Bounds

The result given in Theorem 3.4 gives an asymptotic lower bound on the channel capacity of the LWE/RLWE/MLWE channel.

Definition 3.3. *The smallest achievable block error probability for a channel $\mathcal{C}$ and any code of cardinality M and blocklength n' under maximum likelihood (ML) decoding is $\varepsilon^{\mathcal{C}}_{min}(n', M)$.*

Table 3.5: Code parameters for different Q for NewHope with DFR $< 2^{-216}$, $(n, q, k) = (1024, 12289, 8)$ for coding over 4 blocks

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
2	3	4084	0.9971	4095	4083	0.9968	0.0586
3	13	4021	1.5559	4088	3992	1.9204	0.1130
4	37	3924	1.9160	4095	3933	1.9395	0.1141
5	115	3679	2.0855	4069	3388	1.9206	0.1130
7	829	2277	1.5606	3268	654	0.4482	0.0264

Remark 3.4. *Notice that we used the letter n' to denote the block length rather than n, which specifies the number of coefficients within the polynomials used in this work.*

Assumption 1. *We assume that for the MLWE channel $\mathcal{C}_{\mathsf{MLWE}}$ it holds that the lowest achievable block error probability $\varepsilon_{min}^{C_{\mathsf{MLWE}}}(n', M)$ for fixed codebook size M and blocklength n' is lower bounded by the cardinality of the largest codebook for the marginalized MLWE channel C_{MLWE}^{marg} with the same blocklength and codebook cardinality, i.e.,*

$$\varepsilon_{min}^{C_{\mathsf{MLWE}}}(n', M) \geq \varepsilon_{min}^{C_{\mathsf{MLWE}}^{marg}}(n', M) \ .$$

This is the analog in the finite blocklength regime of the second inequality in equation (3.19)*.*

In the following, we motivate why Assumption 1 is reasonable. First, in the asymptotic setting, the statement holds and has been proved (see Theorem 3.4). Furthermore, this assumption is weaker than the frequently used independence assumption of coefficient failures which indeed has been shown not to hold in general. The construction of BCH codes with their usual encoders and decoders typically does not take into account the channel's memory, i.e., the decoder assumes a memoryless channel, and therefore, we are in a mismatched decoding scenario. That means even though for the channel with memory, it may theoretically be possible to achieve better error performance, due to the suboptimal decoding strategy, the system can have a larger block error probability.

In this section we present results obtained using the normal approximation and the random coding union bound (RCU bound) [PPV10] to get more precise results for the rather short blocklength regime our schemes operate in. Both approximations are given in Chapter 2 in Theorems 2.7 and 2.8, respectively.

While the normal approximation serves as a second order approximation for the achievable rate and is comparably easy to compute, the RCU bound is truly an achievability bound. However, computing the RCU bound is impractical even for moderately complex channels. Hence, we approximated the RCU bound in Fig. 3.7 and Fig. 3.8

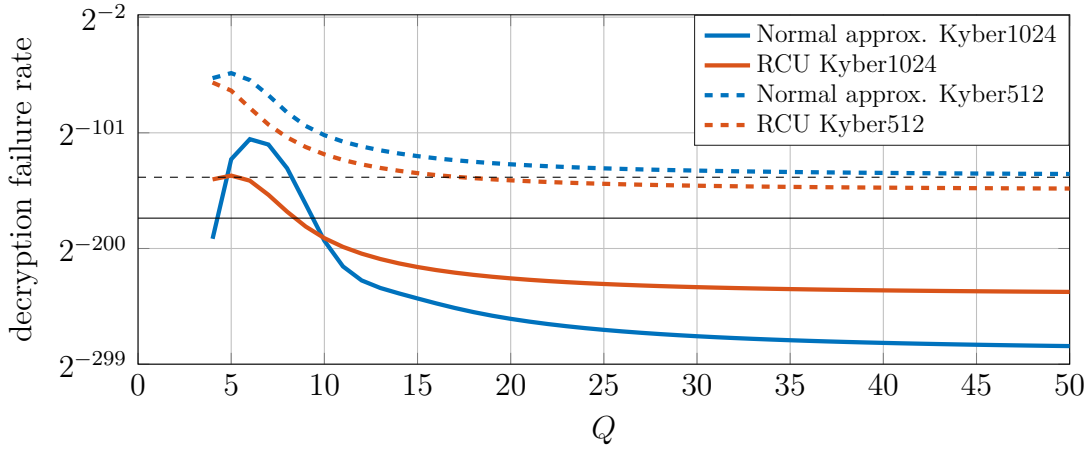

Figure 3.7: RCU bound and normal approximation for the Kyber512 parameter set with $d_u = 7, d_v = 4$ and for the Kyber1024 parameter set with $d_u = 8, d_v = 3$, required DFRs for Kyber512 (black, dashed) and Kyber1024 (black)

using the saddlepoint approximation according to [MF11; FSVVM+18]. We recapitulated the use of the saddlepoint approximation to compute the RCU bound in Section 2.3.1 (Theorem 2.9).

For Kyber512 a DFR of 2^{-139} and for Kyber1024 a DFR of 2^{-174} is necessary according to the specification. Our goal is to share an AES key of 256 Bit. Therefore, we need a bitrate of 1 if we want to share the key using one MLWE block. Fig. 3.7 shows that for Kyber512, it is possible to use ciphertext compression parameters $(d_u, d_v) = (7, 4)$, which corresponds to a reduction in ciphertext size of 25% and for Kyber1024 a reduction of 28% for a single MLWE block. Fig. 3.8 shows that the reduction of the parameter l, which is asymptotically possible, cannot be achieved for a single MLWE block.

3.9 Summary and Open Problems

In this chapter we have shown how to treat RLWE/MLWE-based cryptosystems as communication channels. We have derived lower bounds on the channel capacities for the parameter sets of NewHope and Kyber for their highest proposed security levels. Our results show that enhancing the alphabet size of the channel input Q increases the established lower bound on the channel capacity. Furthermore, we have shown why this effect saturates at a certain point. We have proved why increasing Q does not have a negative effect on the security level of RLWE/MLWE based cryptosystems as long as increasing the DFR is avoided. We have performed the same analysis for the quantized RLWE/MLWE channel. For Kyber the results show that asymptotically a

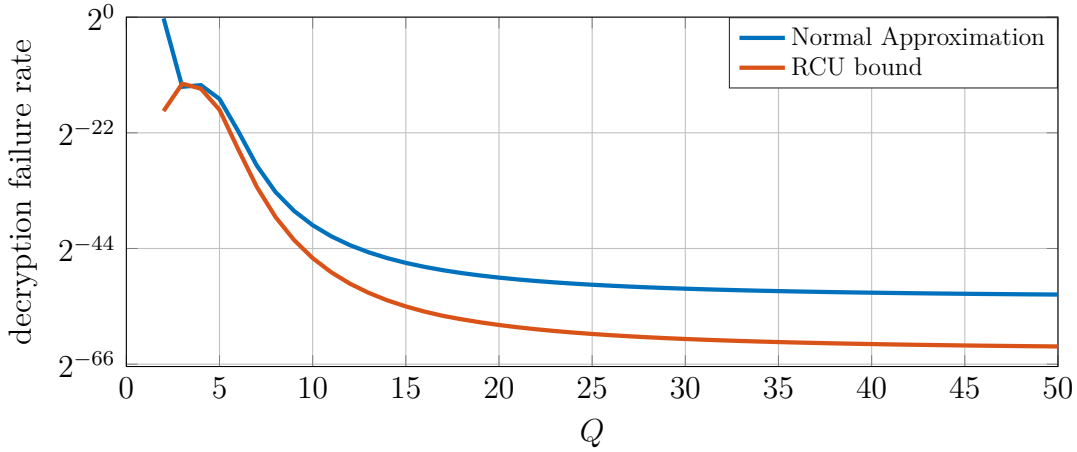

Figure 3.8: RCU bound and normal approximation for Kyber with $l = 3$ and $k = 14$ achieving the security level of the Kyber1024 parameter set

rate increase of almost a factor of 4 is possible while for Newhope an asymptotic rate increase of approximately 10 is possible.

Under the assumption of stochastically independent coefficient failures we have presented achievability results regarding the bitrate based on the Gilbert-Varshamov bound for the parameter sets and required decryption failure rates of NewHope and Kyber. Recall that this bound does not give practical code constructions. Therefore, we have also given bitrates that can be achieved by using practically implementable BCH codes for the same paramter sets. Our results show that we are able to increase the bitrate of NewHope approximately by a factor of 7 and that the rate of Kyber can be increased by a factor of 1.84. Furthermore, we have shown that we can significantly reduce the decryption failure rates for NewHope and Kyber for fixed minimal bitrates of 0.25 and 1, respectively.

Moreover, in the asymptotical setting, we have shown that for Kyber1024 that the secret key and the ciphertext sizes can be reduced by 25%. The public matrix $\boldsymbol{A}$ shrinks by 44% in this case. Furthermore, if one is only concerned about the ciphertext sizes a reduction of 39% and 33% can be achieved for Kyber1024 and Kyber512, respectively. If one only uses a single Kyber encryption to transmit 256 Bits of plaintext to share an AES key it is possible to reduce the ciphertext sizes by 28% and 25%, respectively.

Interesting open problems for this chapter are a better characterization of the stochastic dependence between decryption failures and to integrate this dependency into decoding algorithms to get better results. Furthermore, proving or further justifying Assumption 1 is still an open problem.

Appendix

3.A Conditions of Corollary 3.2 for NewHope

NewHope is specified not to have compression of the ciphertext component $\boldsymbol{u}$. Therefore, in this section we only consider compression of v and its respective compression noise c_{N_v}. This corresponds to the case described in Corollary 3.2.

We distinguish two ways in which ss' can become equal to the zero polynomial, i.e. possibilities such that the Corollary is not applicable. The first one is that either s or s' is equal to zero. Since the error distribution χ_k is known, it is easy write a script computing that this probability is about $2.72 \cdot 10^{-724}$. The other possibility is that $ss' = (x^n + 1)h$, where h denotes an arbitrary non-zero polynomial in $\mathcal{R}_q$. It can be shown that for the parameter set $(n, q, k) = (1024, 12289, 8)$ the polynomial $x^n + 1$ factorizes in linear factors with each linear factor occurring at most once by using a simple sage script. The zeros of the polynomial are not concentrated in a small subinterval of $[0, q-1]$ but rather distributed over the entire interval.

It holds that

$$ss' \equiv 0 \mod (x^n + 1) \Leftrightarrow ss' = (x^n + 1)h$$

for some polynomial $h \in \mathcal{R}_q$. In order for this scenario to occur all roots of $(x^n + 1)$ have to occur at least once in either s or s'. In the following we denote the roots of $(x^n + 1)$ by $\alpha_1, \ldots, \alpha_n$ and consequently

$$s(x) = (x - \alpha_{\pi(1)})(x - \alpha_{\pi(2)}) \ldots (x - \alpha_{\pi(j)})\ a(x) \tag{3.23}$$

$$s'(x) = (x - \alpha_{\pi(j+1)}) \ldots (x - \alpha_{\pi(n)})\ b(x) \tag{3.24}$$

for some $j \in \{1, \ldots, n-1\}$ and an arbitrary permutation π of the set $\{1, \ldots, n\}$.

In contrast to the previous case where we computed the probability that one of the polynomials s or s' is equal to zero for the following analysis we consider them to be sampled from the uniform distribution on $\mathcal{R}_q$. This simplifies our analysis and we justify this methodology by the fact that the roots of $x^n + 1$ are distributed over the entire interval $[0, q-1]$. Therefore, we assume it even to be more likely that within s and s' all roots $\alpha_1, \ldots, \alpha_n$ are contained compared to the case where s and s' are sampled form χ_k.

Due to the uniform sampling of s and s' we can compute the probability that ss' contains all roots by counting the pairs $(s, s') \in \mathcal{R}_q^2$ fulfilling this constraint and

dividing this number by the total number of polynomials in $\mathcal{R}_q^2$ which is q^{2n}.

Proposition 3.2. *The probability that $ss' = (x^n+1)h$, for some non-zero polynomial $h \in \mathcal{R}_q$ is upper bounded by $(2/q)^n$.*

Proof. The polynomials x^n+1 has n roots which are to be distributed to the polynomials s and s'. There are $\binom{n}{j}$ ways to choose j roots of x^n+1 to be roots of s whereas the remaining $n-j$ roots are to be roots of s'. In that case the polynomial $a(x)$ in (3.23) can be chosen arbitrarily from the set of polynomials with degree less than $n-j$. Therefore, there are q^{n-j} possibilities for s. A similar argument shows that there are q^j possibilities for s'. Summing over all possibilities of j we obtain

$$\sum_{j=1}^{n-1} \binom{n}{j} q^{n-j} q^j = q^n(2^n-2) \tag{3.25}$$

where we overcounted for instance the cases where α_1 is a root in both s and s'. Therefore, we get an upper bound on the number of possibilities for s and s' such that $ss' = (x^n+1)h$. Dividing this number by the amount of polynomials in $\mathcal{R}_q^2$ we obtain the desired result. ■

By plugging the parameter set of NewHope into the upper bound of Proposition 3.2 we obtain a value of about $3.9 \cdot 10^{-3880}$. We consider the probabilities of both cases for which $ss' = 0$ to be small enough to consider the conditions of Corollary 3.2 to be fulfilled for the investigated parameter set of NewHope.

3.B Conditions of Lemma 3.2 for Kyber

Recall that for Kyber we have the parameter set $(n, q, k, l) = (256, 3329, 2, 4)$. For this parameter set the polynomial (x^n+1) factors into irreducible polynomials of order 2. To guarantee independence of c_{N_v} from $\boldsymbol{c}_{\boldsymbol{N_u}}$ and the terms generated by the difference noise combined, we require that there exists a pair $(i,j) \in \{1,\dots,l\}^2$ such that $s_i s'_j \neq 0$.

Throughout this section we fix the indices i, j, w. We aim at showing that for these indices indeed the probability that the conditions of Lemma 3.2 are not fulfilled is very small.

Similar to Section 3.A we split our analysis into two cases. For the first case we compute the probability that one of the polynomials s_i, s'_j or s'_w is equal to the zero polynomial. Again this computation can be performed by a simple script which shows that this probability is about 2^{-360} which is way below the desired security level of 256 bit.

The other possibility is that either $s_i s'_j = (x^n+1)h_1$ or $s_i s'_w = (x^n+1)h_2$, where h_1 and h_2 denote arbitrary non-zero polynomials in $\mathcal{R}_q$. It can be shown for the

parameter set $(n, q, k, l) = (256, 3329, 2, 4)$ that the polynomial $x^n + 1$ factors into distinct irreducible polynomials of the form $x^2 - \alpha_r$ with α_r being an element in $\mathbb{Z}_q$. Again this can be checked by using a simple sage script.

Proposition 3.3. *The probability that either $s_i s'_j = (x^n + 1)h_1$ or $s_i s'_w = (x^n + 1)h_2$, where both h_1 and h_2 are non-zero polynomials in $\mathcal{R}_q$ is upper bounded by $2^{n/2+1}/q^n$.*

Proof. As in Section 3.A for the second step we consider the polynomials s_i, s'_j and s'_w to be sampled from the uniform distribution on $\mathcal{R}_q$. We aim at computing the probability that either $s_i s'_j = (x^n + 1)h_1$ or $s_i s'_w = (x^n + 1)h_2$, where h_1 and h_2 are non-zero elements in $\mathcal{R}_q$. By using the union bound and the fact that all elements are sampled independently from the same distribution we have that

$$\begin{aligned} Pr(s_i s'_j = ((x^n + 1)h_1 \vee s_i s'_w = (x^n + 1)h_2) \\ \leq Pr(s_i s'_j = (x^n + 1)h_1) + Pr(s_i s'_w = (x^n + 1)h_2) \\ = 2Pr(s_i s'_j = (x^n + 1)h_1) \ . \end{aligned}$$

Therefore, we just need to show that the upper bound $Pr(s_i s'_j = (x^n+1)h_1) \leq 2^{n/2}/q^n$. As in Section 3.A we investigate that we distribute the aforementioned irreducible polynomials to s_i and s'_j and upper bound the amount of pairs (s_i, s'_j) fulfilling $s_i s'_j = (x^n+1)h_1$ for some non-zero polynomial h_1. The number of such pairs (s_i, s'_j) is upper bounded (same overcounting argument as in Proposition 3.2) by

$$\sum_{p=1}^{\frac{n}{2}-1} \binom{\frac{n}{2}}{p} q^{n-2p} q^{2p} = q^n(2^{n/2} - 2) \leq q^n 2^{n/2} \ .$$

■

By plugging the parameter set of Kyber into the upper bound of Proposition 3.3 we obtain a value of about $1.3 \cdot 10^{-863}$. The probabilities for both analyzed cases are below the security level and therefore, we consider the conditions of Lemma 3.2 to be fulfilled. Practically the likelihood that the required conditions for Lemma 3.2 are not fulfilled are even much lower. We just avoided more complicated combinatorial arguments here.

3.C Results for Frodo and LAC

In this section we present the results for Frodo and LAC analogously to Section 3.4 and Section 3.7 using parameters for their highest respective security level. Most of the results for LAC have already been presented at the ITW 2020 [MPWZ21]. The results Frodo have not been published before and also show that our framework is applicable for standard LWE based schemes. The analysis for LWE based schemes is very similar

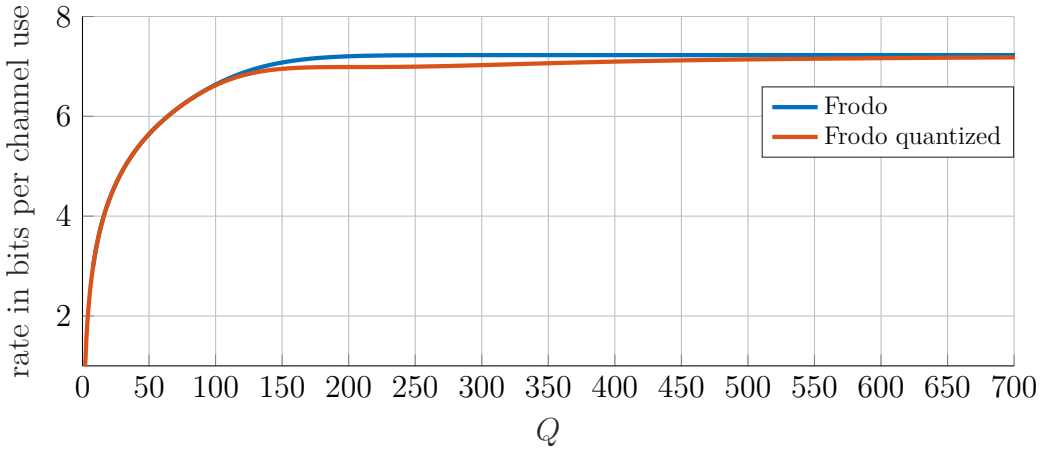

Figure 3.C.1: Lower bound on the capacity of the LWE channel according to Frodo

to MLWE or RLWE based schemes and is therefore omitted. The applicability of the results to Frodo also demonstrates that the error distribution is not restricted to the centered binomial distribution but rather it can be adopted to other distributions (e.g. discrete Gaussian distributions) as well. Dedicated analysis for compression noise like we conducted in Appendix 3.A for NewHope and in Appendix 3.B is unnecessary for LAC and Frodo since within both schemes ciphertext compression is not deployed. For LAC we analyzed the initial submission for Round 1 of the NIST PQC competition because for our framework it is essential that the components of s, e, s', e' need to be sampled independently which is not the case for the Round 2 submission, where the number of $+1$ and -1 elements within the error distribution are fixed. Fig. 3.C.1 and Fig. 3.C.2 show the lower bounds on the achievable rates of Frodo and LAC, respectively. The respective bounds on the plaintext bits per ciphertext bits are presented in Fig. 3.C.3 and Fig. 3.C.4. The results for BCH coding over one ciphertext block coding are given in Tables 3.C.1 and 3.C.2 for Frodo and LAC, respectively. Furthermore, we give results for coding over 4 LWE/RLWE blocks in Tables 3.C.3 and 3.C.4.

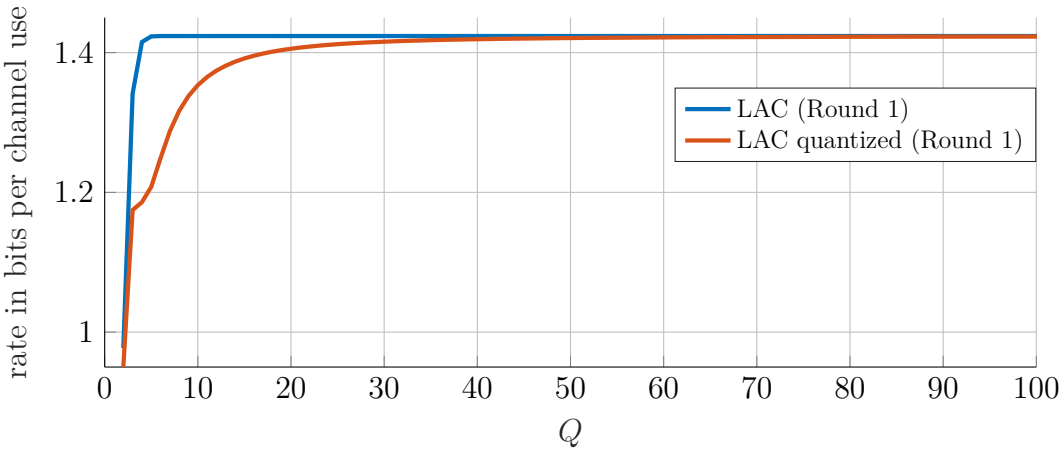

Figure 3.C.2: Lower bound on the capacity of the RLWE channel according to LAC (Round 1 submission)

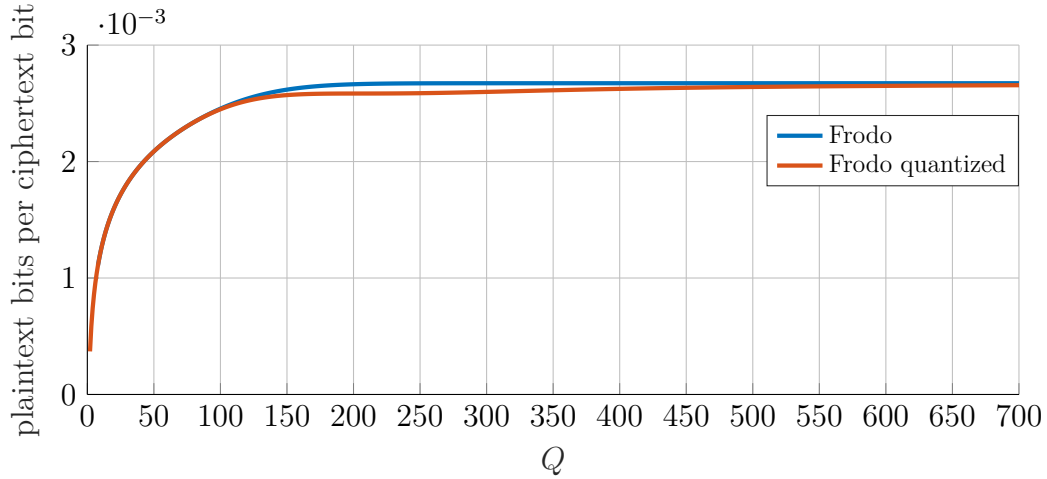

Figure 3.C.3: Lower bounds on the maximal amount of plaintextbits per ciphertextbits for the LWE channels of Frodo

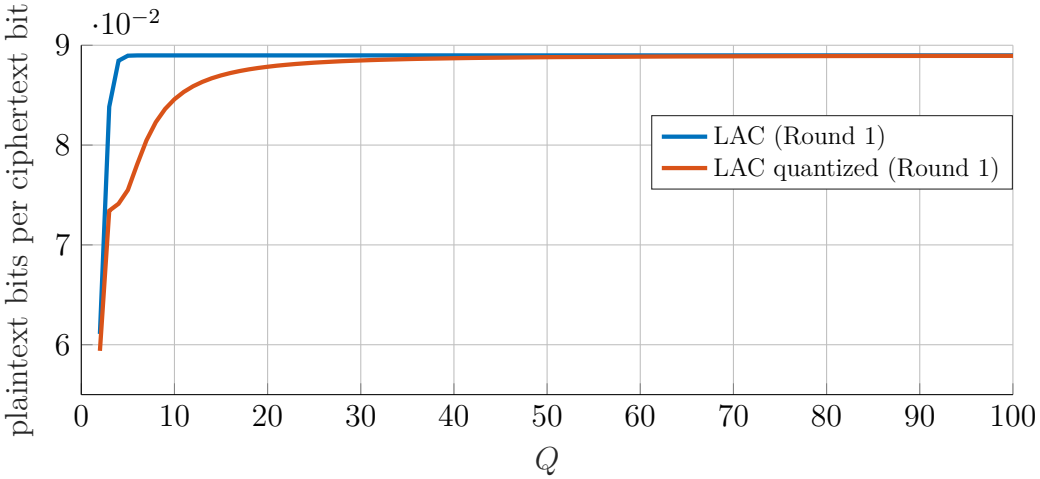

Figure 3.C.4: Lower bounds on the maximal amount of plaintextbits per ciphertextbits for the LWE channels of LAC (Round 1 submission)

Table 3.C.1: Code parameters for different Q for Frodo with DFR $< 2^{-252.5}$, $(n, q, s, l) = (1, 65536, 6, 8)$

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
16	1	–	–	–	–	–	0.00148
17	3	61	3.8959	64	59	3.7681	0.00139
19	3	61	4.0488	60	57	3.7833	0.00140
23	5	57	4.0288	63	52	3.6754	0.00136
27	5	57	4.2348	61	51	3.7891	0.00140
29	7	54	4.0989	60	51	3.8712	0.00143
31	7	54	4.1801	64	55	4.2575	0.00157
32	7	54	4.2188	63	45	3.5156	0.00130

Table 3.C.2: Code parameters for different Q for LAC256 and DFR $< 2^{-115.4}$

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
2	113	521	0.5088	1023	513	0.5010	0.0313
3	373	183	0.2833	728	51	0.0789	0.0049
4	659	24	0.0469	1023	32	0.0469	0.0029

Table 3.C.3: Code parameters for different Q for Frodo with DFR $< 2^{-252.5}$, $(n, q, s, l) = (1, 65536, 6, 8)$ for coding over 4 blocks

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
19	3	253	4.1981	254	250	4.1484	0.00153
27	5	248	4.6063	242	231	4.2905	0.00159
29	7	244	4.6303	240	226	4.2887	0.00159
31	7	244	4.7220	256	232	4.4897	0.00166
32	7	244	4.7656	255	231	4.5117	0.00167
47	17	226	4.9037	255	211	4.5782	0.00169
57	27	211	4.8076	250	201	4.5797	0.00169
67	39	194	4.5970	255	143	3.3885	0.00125

Table 3.C.4: Code parameters for different Q for LAC256 and DFR $< 2^{-115.4}$ for coding over 4 blocks

Q	d	k_{GV}	R_{GV}	n_{BCH}	k_{BCH}	R_{BCH}	plain/cipher
2	211	521	0.5088	4095	2895	0.7068	0.0442

4

Information Theoretic Analysis of PUF-Based Tamper Protection

Abstract

PUF-based approaches enable tamper protection for high-assurance devices without needing a continuous power supply. Several methods for PUF-based tamper protection have been proposed together with practical quantization and error correction schemes. In this chapter we take a step back to analyze theoretical properties and limits. We apply zero leakage output quantization to existing quantization schemes and minimize the reconstruction error under zero leakage. We apply wiretap coding within a helper data algorithm to enable key reconstruction for the legitimate user while making reconstruction impossible for the attacker models considered in this chapter. We present lower bounds on the achievable key rates depending on the attacker's capabilities in the asymptotic and finite blocklength regime to give fundamental security guarantees even though the attacker gets partial information about the PUF response. Furthermore, we present converse bounds on the number of PUF cells. Our results show for example that for a practical scenario one needs at least 459 PUF cells using 3 bit quantization to achieve a security level of 128 bit.

The results in this chapter are joint work with Matthias Hiller. We have submitted a paper containing several results presented in this chapter to the IACR Transactions on Cryptographic Hardware and Embedded Systems (TCHES) and received a Major Revision. We are planning to resubmit the work for one of the next two deadlines of the same conference.

4.1 Introduction

Physical Unclonable Functions (PUFs) evaluate physical properties of devices to obtain unique identifiers of electronic devices and physical roots of trust for cryptographic keys. Furthermore, PUFs can be used as a foundation for tamper protection technology that facilitates to validate the physical integrity of an embedded system after its power-up. All approaches have in common that minuscule manufacturing variations

within physical objects, or mostly electronic components, are evaluated to generate a device-unique output. While there are several works on the assessment of the entropy or randomness of PUFs [MGS13; WGP18; FWHP23] and the leakage through the published helper data necessary within the reconstruction phase, e.g. [DGSV15; DGV+16], we are currently lacking a theoretical model to quantify the security in the light of a physical attacker who destroys parts of the PUF response to read out the remainder.

Going from silicon PUFs, e.g. SRAM PUFs, to system-level PUFs facilitates to incorporate tamper protection capabilities to protect an entire embedded device with components that cannot resist advanced physical attacks on their own, such as processors or FPGAs, and their communication. This reduces the attack surface from several individual vulnerabilities, e.g., against laser or EM fault injection and EM or optical side channel analysis to the attack resistance of the surrounding foil. We also consider it impossible to perform power side-channels since the attacker can only access the power supply of the entire printed circuit board and has no direct access to the power supply of the chip.

For the remainder of this chapter, we will refer to the PUF-based tamper protection foil proposed by Immler *et al.* [IOK+18; ION+19]. However, the generic results of this work can be adapted and applied to other PUF types as well. Examples are the coating PUF [TSS+06b] and the polymer waveguide PUF [VWN+16; GGV17]. The tamper protection is based on a foil that is wrapped around a Printed Circuit Board (PCB), or a cover that is placed on top. This foil consists of a mesh of electrodes, leading to a large number of capacitances that can be monitored by a mixed signal circuit from within the protected area. It evaluates the capacitive coupling between electrodes to derive the cryptographic key and to validate the physical integrity of the system and performs run-time tamper detection during operation to protect the system.

One of the critical attack vectors, considered during the evaluation of hardware devices with security boxes [SI23], is that an attacker penetrates the foil with a small needle or drill and accesses internal signals. If the required drill diameter is sufficiently large, major changes occur in the capacitance measurements of a significant portion of the foil, leading to an incorrectly reconstructed PUF response during the reconstruction phase. Therefore, the secret cannot be uncovered by an attacker, as discussed, e.g., in [GXKF22a]. As the foil's PUF values may also change over time due to noise, aging, and varying environmental conditions such as temperature or humidity, an error-correcting code is implemented in the system to compensate for those effects to ensure that the correct cryptographic key is derived with a probability $> 1 - 10^{-6}$ or even $> 1 - 10^{-9}$.

Our goal is to analyze the resulting wiretap channel [Wyn75; CK78] between the enrollment and reconstruction phase of the legitimate user as well as the reconstruction phase of the attacker from an information theoretical point of view. We establish lower

bounds on the secrecy capacities of the resulting channels as well as finite blocklength achievability and converse bounds on the maximal achievable secrecy rate, making our results relevant in practice as they provide benchmarks for implementations by quantifying the distance of a practical implementation to the theoretical limit.

4.1.1 Main Results

The main results presented in this chapter are:

- Information theoretical channel model including zero leakage helper data generation for physical tampering with PUFs
- Asymptotic results for lower bounds on the channel capacity of the resulting PUF-channel under different attack scenarios
- Finite blocklength achievability and converse results on the number of required capacitances to achieve a predefined security level in two attack scenarios
- Proof that previously used helper data schemes do not achieve required security levels without leaking information about the secret via the helper data
- Quantitative results that demonstrate that a 128 bit security level is achievable with 1400 and PUF cells for 18% and 36% erasure probability for digital and analog attacker, respectively

4.1.2 Outline

Section 4.2 gives a brief overview over related work. Section 4.3 gives some background information on the foil PUF and introduces the resulting channel model. In Section 4.4, we obtain results on the secret key capacity of the HDA for digital and analog attacker. Section 4.5 investigates secret sharing using common randomness using one-way communication for finite blocklength. It serves as a foundation to analyze the HDA performance with respect to the required amount of capacitive PUF cells for the foil PUF presented in Section 4.6. In Section 4.7 we use the converse result on the necessary amount of PUF cells to show that a given implementation has to either leak via the helper data or be insecure by other means. Section 4.8 sums up the results and states open problems.

4.2 State of the Art

While some silicon PUFs such as the SRAM or arbiter PUF directly output digital information, other silicon PUFs, like the Ring-Oscillator or TERO PUF, and in particular non-silicon PUFs output analog, or finely quantized digital data. They all have

in common, that they undergo a long processing chain, until a cryptographic key is output (see Section 2.7.3).

4.2.1 PUF-Based Tamper Protection

In the past, tamper protection was implemented through a continuously powered detection mechanism, e.g. in tamper-responsive envelopes and covers that wrap or cover the structure to be protected [OI18]. A physical measure such as electrical resistance is measured over a protective structure and triggers an alarm as soon as the measurement exceeds a threshold to erase sensitive information, stored e.g. in battery-backed memory. While the device is only active during a fraction of the time, the protective measures need to be active for the entire life-time of the device, after it leaves a trusted manufacturing site. PUF-based tamper protection promises to increase sensitivity and to ease the handling during operation of the devices. Thus PUFs can contribute to an easy-to-handle and future-proof tamper protection technology.

Over the last 20 years, different measures were taken to combine PUFs and tamper protection. An early type is the coating PUF, where a protective coating is spread over an IC and evaluated from its inside to detect physical tampering when the coating is penetrated [SMKT06; TSS+06a]. In addition to electrical measurements, also optical approaches [EFK+12; VNK+15] or propagation behavior of radio waves [ZHS18] were proposed.

In this work, we use PUF-based tamper protection realized by foils and covers, as also proposed in [IOK+18; Imm19; ION+19; GOFK21]. This scenario is depicted in Figure 4.2.1. The foil is wrapped around a PCB with mounted electronics components to be protected and consists of two structured electrode layers (blue and red) and two electrical shielding layers (black). The capacitive coupling between the electrode layers is measured to obtain the PUF response [OIHS18]. In addition, faster mechanisms for run-time tamper protection are included.

The capacitance values are subject to measurement noise, temperature and other environmental effects, as discussed based on practical values, e.g. in [GXKF22b]. A broad range of deterministic effects can be compensated with linear or also higher order reference points of fits [OIHS18; GXKF22b; RFB+23], whereas Gaussian measurement noise remains in all cases. We will focus on this fundamental issue using a wiretap channel in this work and refer to the compound wiretap channel for considering multiple environmental conditions [LKP09].

4.2.2 Quantization

All analog PUFs evaluated on embedded devices need to quantize the digitized PUF data into a finite alphabet that is input into the error correcting code (see Section 2.7.5). Typically, public helper data that references the distance and direction to the next interval border is stored to move data point away from decision borders right

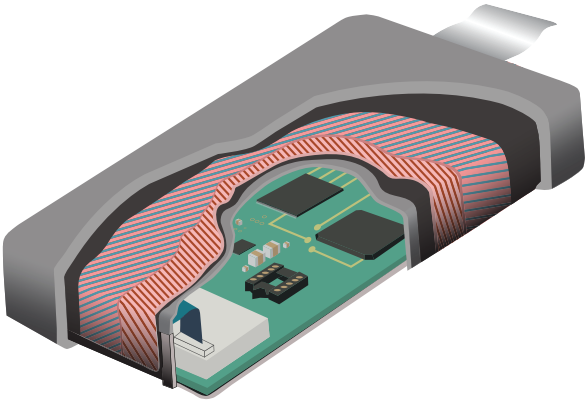

Figure 4.2.1: Sketch of PUF-based envelope [IOK+18] with two conducting electrode layers in red and blue, and two shielding layers in black (figure copied with permission of Matthias Hiller)

into the middle of the quantization interval. So far, work on PUFs typically either uses equiprobable or equidistant quantization. As shown in Section 2.7.6 it is possible though to use arbitrary input quantization while still generating helper data that is not leaking any information about the secret (zero-leakage helper data).

Equiprobable Quantization The range of possible output values of the PUF is split into d intervals with the same probability such that all indices as sampled with the same probability. This is favorable from a security point of view, as iid PUF values are mapped into uniform data in the finite alphabet. However this comes with two downsides: First, the common helper data generation bringing the expected value during reconstruction into the center of the interval leaks information about the secret, as the helper data pointers of different intervals have different distributions [IHKS16]. Later in this chapter we will introduce a method to obtain zero leakage helper data such that this problem is mitigated. Second, the decision borders in the center are rather narrow, so that the values are subject to a higher error rate. The wide intervals in the tails of the distribution also have a decreased sensitivity for physical tampering.

Equiprobable quantization can be used if the requirement for uniformity exceeds the requirements for tamper detection sensitivity [GXKF22b].

Equidistant Quantization In contrast, equidistant quantization samples the analog value mapping it to intervals of the same size. This has the advantage that additive noise effects all values in the same way and error probabilities are constant with respect to the PUF value. Also, only a negligible leakage can be observed [IHKS16] using the aforementioned helper data pointers. With the later on introduced zero leakage helper data algorithms this is less of a factor. As a downside, this comes at the expense of a heavy bias as the intervals differ in probability. This can be mitigated through

variable-length encoding at the expense of a limited selection of code classes for the later ECC [IHL+18]. As shown in [BDH+10], also higher-dimensional structures can be used for embedding secret data, which adds more degrees of freedom.

4.2.3 Error Correction

Aside from the helper data used to perform better output quantization during reconstruction (denoted by W^n in Section 2.7.3) additional helper data $\widetilde{W}$ is generated to link the PUF response to a codeword of an error correcting code. This can be done either by linear schemes that generate the link through linear dependencies such as syndromes or XORs or pointers that refer to parts of the PUF response [HKS20].

In any case, an ECC is used to reduce the key error probability e.g. down to 10^{-6} or 10^{-9} to generate reliable keys. This can be done with standard codes such as BCH, Reed-Solomon or Convolutional codes [MS77a], or newer code classes such as limited magnitude codes [IU19] or Polar codes [CIW+17; GXKF22b].

In addition, wiretap codes can be used either for leakage prevention [HO17] or attack prevention [GXKF22b].

Over the last decades, several schemes have been proposed and implemented to address the design of the error correcting codes (ECCs) and helper data, see e.g. [DGSV15; HKS20]. In the following, we focus on the generation of input and output quantizers as well as funcamental theoretical limits on the amount of required PUF cells. Designers can then benchmark their implementations against those fundamental limits. The construction of a practical wiretap code and considerations for the leakage of the helper data connecting the PUF responses and the coding scheme are out of the scope of this thesis.

4.3 PUF-Based Tamper Protection Foil

Going from the pure PUF-based key generation scheme to tamper protection brings additional requirements for the PUF. The combination of enrollment phase and reconstruction phase using HDAs as proposed in Section 2.7.5 for the combination of legitimate user and a physical attacker is related to secure communication over a discrete memoryless degraded wiretap channel or a discrete memoryless wiretap channel depending on the chosen attacker models that are specified in the following.

4.3.1 Attacker Model

Aside from the standard attack vectors of key generation schemes using analog PUFs like helper data leakage in addition the physical tampering aims at obtaining secret information and this needs to be prevented by the designer.

Considering all attack vectors, the remaining security of the system requires to exceed a specified minimum security level such as 120 bit, as currently recommended by the BSI [BSI24]. In this work we chose to investigate 128, 192 and 256 Bit security levels as common in many cryptographic applications.

Leakage through Helper Data Since early approaches like the fuzzy commitment or fuzzy extractor [DRS04b], helper data leakage is a topic to be considered when designing secure key generation with noisy secrets, and in particular for PUFs [DGSV15] when imperfections come into play [DGV+16]. For the remainder of the this chapter, we assume a random number with full entropy for selecting the codeword and refer to the mentioned related works to design helper data schemes that avoid leakage through $\widetilde{W}^n$ by the helper data algorithm. In addition to $\widetilde{W}^n$, the quantization helper data W^n needs to be considered in addition for analog PUFs and we are using the zero leakage helper data generation proposed in Section 2.7.6 for W^n.

Physical Tampering As physical tamper protection aims to withstand attackers being able to use sophisticated tools, a wide range of attacks needs to be considered [Wei00; Imm19; GSHO21; SI23], where a special emphasis is put to physical drilling, that affects only very small areas.

As reference, for the system proposed in [IOK+18], drilling with a conventional drill bit with diameter $> 300\mu m$ fully destroys one electrode in both layers, such that 23 out of 128, or 18% of the capacitance values are destroyed. Attacks during operation can be detected with the integrated run-time tamper detection measures. Attacks on the powered-off device are more challenging from a theoretical point of view. For identical layouts, we need to assume that the attacker knows the position of the destroyed PUF cells within the PUF response, which we model as erasures in the proposed coding schemes.

A neuralgic point for an attack lies between the measurement circuit and the key generation, as the attacker can obtain the digitized low-noise values from tapping a single wire, before the values are interpreted in the embedded key management system and the attack detection triggers. The attacker can only perform a single measurement of the PUF in this case because after this measurement the chip will notice that the foil has been attacked and burns a fuse bit, eliminating the possibility for further measurements. In the following this attacker is referred to as the "digital attacker" for brevity.

In another scenario, the attacker performs more advanced analog measurements, i.e. we assume he affords better measurement equipment. We call this kind of attacker the "analog attacker" for short. We make the very conservative assumption the attacker can measure with infinite precision, i.e. his equipment has infinitely fine quantization steps and furthermore he can perform an unlimited number of measurements. Due to the unlimited amount of measurements with uncorrelated measurement noise the

attacker is able to apply post-processing to effectively eliminate the measurement noise. However, in this scenario the attacker is required to drill multiple holes to tap multiple analog wires, potentially with a larger wire diameter instead of only one small hole to probe a digital signal. We will show in Section 4.4 that this attack is indeed problematic, especially for larger field sizes and hence propose to use countermeasures on the hardware level eliminating the possibility for an attacker to perform those advanced measurements. A profound reasoning for the amount of PUF cells being destroyed by this attack is out of scope of this work. It is reasonable though to assume that a significant extra proportion of the foil needs to be destroyed compared to the digital attacker to mount such an attack. In our examples we used twice the amount of destroyed capacitances compared to the digital attacker.

Within the helper data algorithm the ECC enables recovering the key from an erroneous PUF measurement as long as the number of errors is sufficiently small. Designers therefore need to take care that an attacker cannot retrieve information about the secret key using the redundancy inflicted by the ECC even though the attacker has to destroy a fraction of the PUF cells to measure the PUF. Investigating fundamental limits of this problem is the major contribution of this chapter.

In addition to the possibility of wiretapping internal signals, the cable connections through the foil that could reveal timing or global power side channels need to be taken into account. This topic will not be considered in the scope of this thesis and can be addressed individually when designing the host system.

4.3.2 Enrollment and Reconstruction Phase for the Foil PUF, Legitimate User

The PUF considered throughout this chapter is established by a foil consisting of electrodes forming a mesh of capacitances. The purpose of this PUF is twofold. First it is used for storing a cryptographic key in a secure manner and secondly it shall protect the components within its inside. An attacker trying to perform a side channel attack is required to measure parameters related to the implementation and the foil prevents him from accessing critical hardware. As composition of physical variations, the capacitances can be modelled as Gaussian random variables [IOK+18]. While the work in [GXKF22a] focuses on the reliability side with a strong focus on the specific implication and its measurements, this work aims at quantifying the information theoretic security limits of this type of PUF.

The distribution of the differentially evaluated PUF response for a single cell follows a Gaussian of zero mean and variance σ_P^2, i.e. $X \sim \mathcal{N}(0, \sigma_P^2)$. We assume that the PUF values for the individual cells to be independently identically distributed (iid). We denote the output one PUF cell during the enrollment phase by the random variable X. To obtain the secret S for this cell, the measured data is quantized (see Section 2.7.5). The generation of the helper data $W$of a PUF is peculiar because not

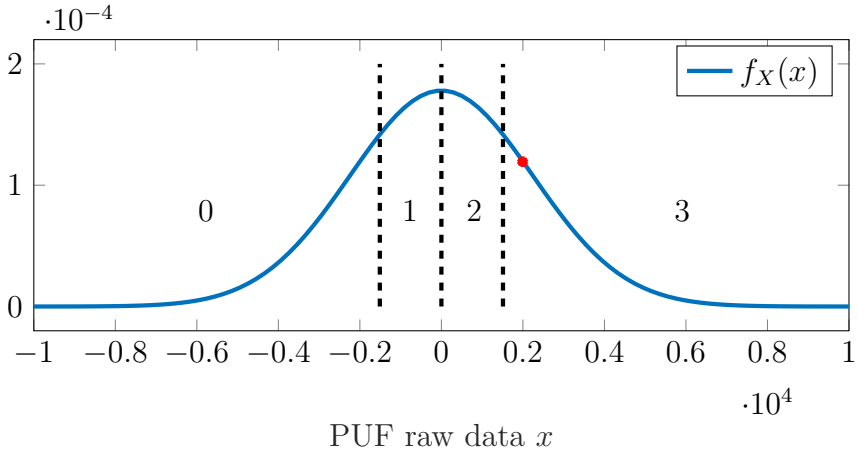

Figure 4.3.1: Enrollment phase

only shall it be of value within the reconstruction process but have the zero leakage property (Definition 2.46). Zero-leakage quantization helper data generation has already been discussed in Section 2.7.6 and we follow this approach throughout this work. In Fig. 4.3.1 the pdf of a single PUF cell is depicted. The red dot represents the realization of X. In this example we used 2 bit quantization to obtain the value of the secret value of the PUF, denoted by S. The dashed lines represent the borders of the quantizer and the numbers inside the intervals represent the respective values of S, i.e. in our example $S = 3$. The location of X within the respective interval is then used to derive the helper data W which is later on used to find an estimate for S during the reconstruction phase.

The PUF response during the reconstruction phase is modelled as the PUF response during the enrolment phase X perturbed by additive Gaussian noise with variance σ_N^2. We denote this output by the random variable $Y = X + N$ with $N \sim \mathcal{N}(0, \sigma_N^2)$. During the reconstruction phase, Y is used in combination with the helper data W to output an estimate for S, in the following denoted as $\widetilde{S}$. Throughout this work we use $\sigma_P = 2241$ and $\sigma_N = 129$ which is consistent with the results in [GXKF22b]. The output quantizer Q_{out} depends on the quantization helper data W and has been specified in Section 2.7.6. We denote the output of this quantizer by $\widetilde{S}$.

The conditional distribution of the PUF response during the reconstruction phase for the example in Fig. 4.3.1 is shown in Fig. 4.3.2. Notice that the quantization intervals during the reconstruction phase have been shifted compared to the enrollment phase. This is due to the utilization of the helper data W. By utilizing the helper data the distance between the value during the enrollment phase (the red dot) and the relevant quantization boundary for the error has increased, leading to a lower reconstruction failure probability.

Our next goal is to secure the HDA for the foil PUF against the attack scenarios

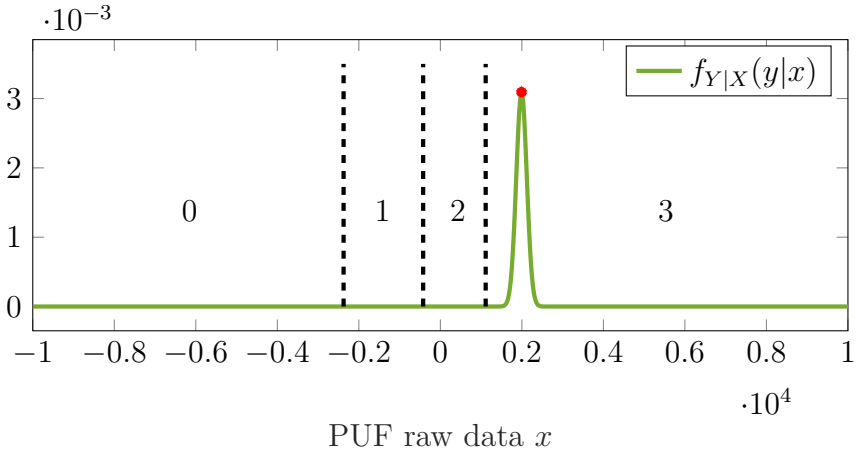

Figure 4.3.2: Reconstruction phase

mentioned in Section 4.3.1. In Section 2.7.4 a connection between secret sharing using common randomness and HDAs for PUFs. However, it has also been pointed out that the secret sharing protocol used to establish Theorem 2.11 requires two-way communication, thereby making it unapplicable in the PUF setting. Hence, in the following we investigate secret sharing using common randomness using one-way communication in the finite blocklength regime.

4.4 Secret Key Capacities for the Foil PUF

In this section we establish the connection between enrollment and reconstruction phase for the legitimate user combined with the attacker models introduced in Subsection 4.3.1 and the secret key generation using correlated sources. The enrollment and reconstruction phase during normal operating conditions as well as for the two mentioned attack scenarios are depicted in Fig. 4.4.1.

We first tackle the problem of determining the maximal code rate of the ECC for a HDA providing security against the attacks described in Section 4.3.1. Essentially this code rate is related to the hardware complexity and the security level of the PUF as outlined in Section 2.7.3.

Updating Fig. 2.4 to also include the attacker models discussed in Section 4.3.1 we arrive at the block diagram shown in Fig. 4.4.1. Depending on whether we exclude the analog attacker by preventing the necessary measurements on a hardware level as proposed in Section 4.3.1 only the variables connected to the decoder for the digital attacker are of concern or also the parts associated with the analog attacker. To keep the analysis of the model simple, we take the approach of examining the scenario where only the digital attacker needs to be considered first. The changes necessary to also

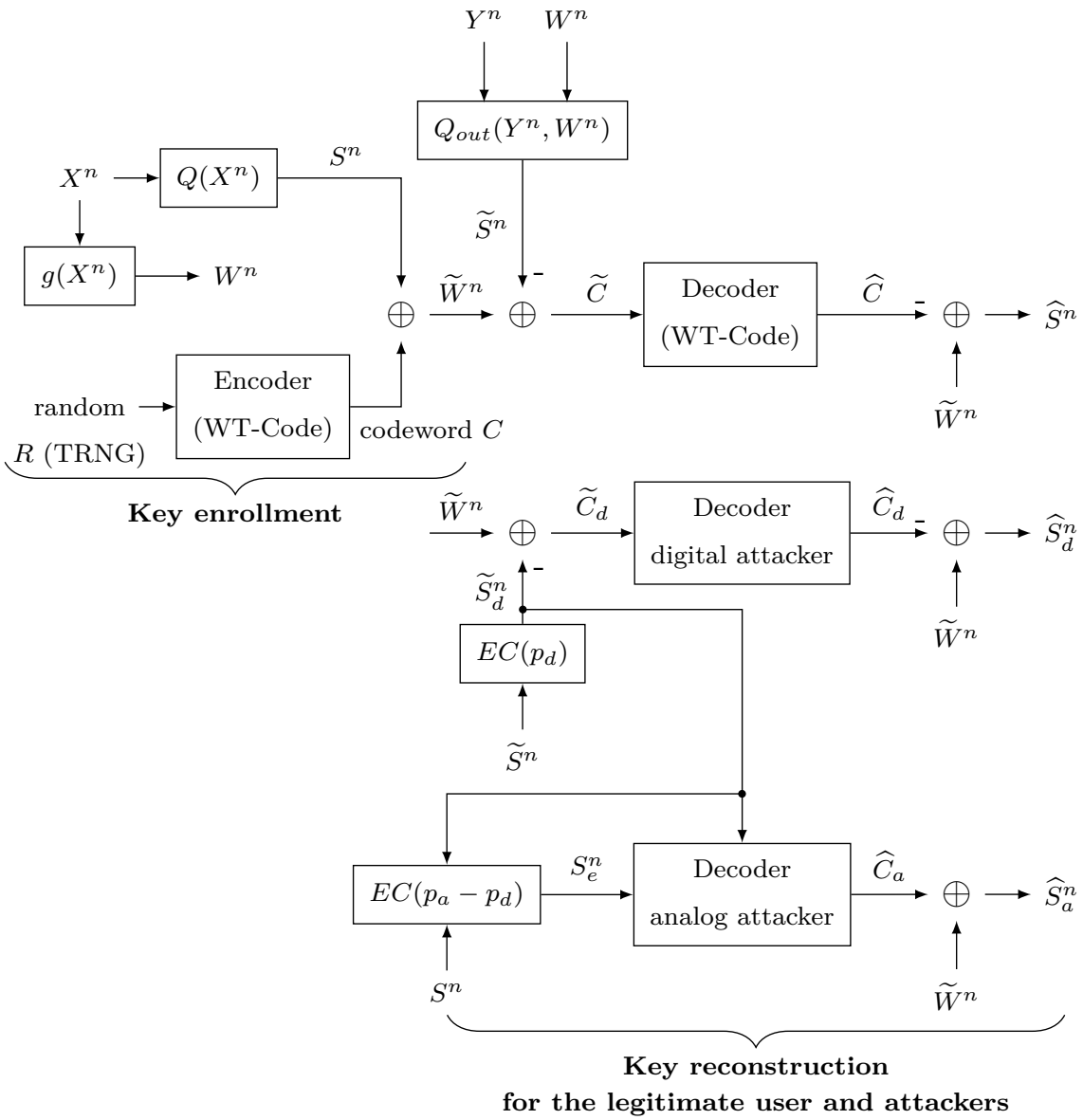

Figure 4.4.1: Key enrollment and key reconstruction for legitimate user, digital and analog attacker

integrate the analog measurements of the PUF then build from there.

We observe that for the reconstruction phase the legitimate user measures Y^n and utilizing the quantization helper data W^n obtains an estimate for S^n referred to as $\widetilde{S}^n$. In contrast via the probing the digital measurement line the attacker obtains the same measurement with additional erasures $\widetilde{S}^n_d$. By a similar argument to the one in Section 2.7.4 we have that the resulting helper data scheme's task is to implement a

secret sharing algorithm using a source of common randomness. The source of common randomness is formed by the joint probability distribution $P_{S,\widetilde{S},\widetilde{S}_d}$. In the secret sharing problem, the first terminal Alice gets access to S^n and wants to agree on a key with Bob who gets $\widetilde{S}^n$, while the eavesdropper Eve obtains $\widetilde{S}_d^n$. As described in Section 2.2.2 Alice and Bob next exchange messages over a public channel and aim to agree on a secret key in a secure manner, i.e. in a way such that it is impossible for Eve to obtain information about the key. Interpreting the HDA as a way to integrate a secret sharing protocol, the information shared over the public channel are the quantization helper data W^n and the reconstruction helper data $\widetilde{W}^n$. Alice's and Bob's goal is to agree on a key which is as large as possible, thereby maximizing the entropy of the key for some arbitrary but fixed n. Results on the secret key capacity (asymptotic setting as $n \to \infty$) have been recapitulated in Section 2.2.2 (Theorem 2.6) and applying these results to the HDA scheme in Fig. 4.4.1 leads to the result in Theorem 4.1.

Definition 4.1. *We consider an HDA with q input quantization levels and an attacker trying to obtain the secret established during the enrollment process. The key capacity $C^q_{key,attacker}$ in this setup is the maximal asymptotic code rate for which a wiretap code exists such that the block error probability for the legitimate user is arbitrarily small and the system achieves an arbitrarily high secrecy level as the blocklength n goes to infinity.*

Theorem 4.1. *The key capacity $C^q_{key,dig}$ for the foil PUF and the digital attacker is optimal in the sense that it enables a code rate for the ECC that is equal to the secret key capacity of the secret sharing problem with common randomness specified by the source $P_{S,\widetilde{S},\widetilde{S}_d}$. More specifically, the code rate $C^q_{key,dig}$ is defined by*

$$C^q_{key,dig} := \max_{input\ quantizer} I(S;\widetilde{S}|W)p_d \ . \tag{4.1}$$

Proof. First of all it is obvious that it is impossible to achieve a rate higher than the secret key capacity of the secret sharing problem using common randomness specified by the joint distribution $P_{S,\widetilde{S},\widetilde{S}_d}$.

For the achievability consider the codewords of the ECC to be sampled randomly and iid from the uniform distribution over the range of S, denoted by $\mathcal{S}$. The HDA in Fig. 4.4.1 precisely implements the construction outlined in the achievability proof sketch of Theorem 2.6. Hence, due to the fact that $\widetilde{S}_e$ is defined to be the output of an erasure channel having erasure probability p_d.

By the proof sketch for the achievability of Theorem 2.6 and by identifying $S \equiv X$, $Y \equiv \widetilde{S}$ and $Z \equiv \widetilde{S}_d$, we have according to Theorem 2.6

$$\begin{aligned}\widetilde{C}_S &= I(X;Y) - I(X;Z) = I(S;\widetilde{S}|W) - I(S;\widetilde{S}_e|W) \\ &= I(S;\widetilde{S}|W) - I(S;\widetilde{S}|W)(1-p_d) = I(S;\widetilde{S}|W)p_d \ .\end{aligned} \tag{4.2}$$

Furthermore, compared to the secret sharing using common randomness problem introduced in Section 2.2.2 we have the additional freedom of choosing the input quantizer. This extra degree of freedom can be exploited since Equation (4.2) is valid for any input quantizer Equation (4.1) follows, completing the proof. ■

In order to compute this capacity the first step is to analyze the relation between S and $\widetilde{S}$ conditioned on the knowledge of the helper data W. The distribution of S follows from the distribution of the PUF during the enrollment phase, i.e. from the distribution of X, and from the choice of the input quantizer. It is complicated to perform the maximization in Theorem 4.1 because the channel matrix is changing according to the input quantizer as its choice influences the output quantizer (see Remark 2.10). The following observation is helpful though.

Lemma 4.1. *For the channel resulting in the concatenation of enrolment and reconstruction phase at the legitimate user it holds that*

$$I(S;\widetilde{S}|W) = I(X;\widetilde{S}|W) \tag{4.3}$$

Proof. By using the chain rule of mutual information it holds that

$$\begin{aligned} I(X,S;\widetilde{S}|W) &= I(S;\widetilde{S}|W) + I(X;\widetilde{S}|W,S) && (4.4)\\ &= I(X;\widetilde{S}|W) + I(S;\widetilde{S}|W,X)\ . && (4.5) \end{aligned}$$

It holds that $I(X;\widetilde{S}|W,S) = 0$ and $I(S;\widetilde{S}|W,X) = 0$ because (W,S) uniquely determines X and X determines S and the statement of the Lemma follows. ■

Since Lemma 4.1 shows that we can focus on $I(X;\widetilde{S}|W)$ and furthermore

$$\begin{aligned} I(X;\widetilde{S}|W) &= H(\widetilde{S}|W) - H(\widetilde{S}|W,X) && (4.6)\\ &= H(\widetilde{S}|W) - H(\widetilde{S}|X)\ , && (4.7) \end{aligned}$$

where the last equality follows because W is a function of X. We observe that the quantity that we are interested in is the difference between the uncertainty of $\widetilde{S}$ given the publicly available helper data W compared to the uncertainty of $\widetilde{S}$ given the private PUF response during the enrollment phase X.

Next we deal with the additional analog attacker. This attacker not only gets access to the digitized measurement of the internal circuitry of the PUF after the erasure channel $EC(p_d)$ but also to an analog measurement of the foil PUF passed through another erasure channel. This channel inflicts an additional fraction of $(p_a - p_d)$ errors into the positions that have not yet been erased by the hole necessary for the digital measurement. Hence, this erasure channel needs to access which cells have been erased by the digital attacker and this is reflected in the additional input $\widetilde{S}_d^n$. Its only purpose is to declare which cells have already been erased in this context.

The attacker's decoder however can utilize the information that $\widetilde{S}_d^n$ provides. This information is only useful for reconstructing the values that have been erased by the additional erasures because the number of measurements for the analog attacker is unlimited and hence unerased symbols are perfect reconstructions of the enrollment value S^n.

Theorem 4.2. *For the key capacity $C^q_{key,ana}$ of the foil PUF and the analog attacker it holds that*

$$\max_{input\ quantizer} I(S;\widetilde{S}|W)(1-p_a+p_d)-H(S)(1-p_a) \leq C^q_{key,ana} \leq \max_{input\ quantizer} I(S;\widetilde{S}|W)p_d \ . \tag{4.8}$$

Proof. We first prove the lower bound. Using Theorem 2.6 we have that

$$C^q_{key} \geq I(S;\widetilde{S}|W) - I(S;\widetilde{S}_d,\widetilde{S}_a|W) \ . \tag{4.9}$$

By the definition of mutual information it holds that

$$I(S;\widetilde{S}_d,\widetilde{S}_a|W) = H(S) - H(S|\widetilde{S}_d,\widetilde{S}_a,W) \tag{4.10}$$

because $H(S|W) = H(S)$ due to the zero leakage condition and furthermore we observe that

$$\begin{aligned} H(S|\widetilde{S}_d,\widetilde{S}_a,W) &= H(S|\widetilde{S}_d=0,\widetilde{S}_a=E,W)P_{\widetilde{S}_d}(0)(p_a-p_d) \\ &\quad + \dots \\ &\quad + H(S|\widetilde{S}_d=|\mathcal{S}|,\widetilde{S}_a=E,W)P_{\widetilde{S}_d}(|\mathcal{S}|)(p_a-p_d) \\ &\quad + H(S|\widetilde{S}_d=E,\widetilde{S}_a=E,W)\,p_d \\ &= H(S|\widetilde{S},W)(p_a-p_d)+H(S)p_d \ . \end{aligned} \tag{4.11}$$

Plugging these simplifications into the right hand side of (4.9), we have

$$H(S)p_d - H(S|\widetilde{S},W)(1-p_a+p_d) \tag{4.12}$$

which is equivalent to the lower bound in the theorem after some simple algebraic steps and maximizing over the choice of the input quantizer.

The upper bound follows trivially because the analog attacker is stronger than the digital attacker. However, we also show that the upper bound cannot be trivially improved by the well known upper bound

$$I(S;\widetilde{S}|\widetilde{S}_d,\widetilde{S}_a,W) = H(S|\widetilde{S}_d,\widetilde{S}_a,W) - H(S|\widetilde{S},\widetilde{S}_d,\widetilde{S}_a,W) \ . \tag{4.13}$$

The first entropy in Equation (4.13) has already been simplified within this proof.

Table 4.4.1: Achievable asymptotic rates for the PUF-channel for weak and strong attackers, different input quantization alphabets and quantization strategies, erasure probability $p_d = 0.1$ for the digital attacker and $p_a = 0.2$ for the analog attacker, $\sigma_P = 2241, \sigma_N = 129$

quantizer levels	equidist. digital	equidist. analog	equiprob. digital	equiprob. analog	optimized digital	optimized analog
2	0.1	0.1	0.1	0.1	0.1	0.1
4	0.117	0.117	0.2	0.2	0.2	0.2
8	0.196	0.196	0.298	0.282	0.299	0.294
16	0.291	0.291	0.356	0	0.363	0.327
32	0.376	0.269	0.382	0	0.382	0.311
64	0.389	0	0.398	0	0.398	0
128	0.404	0	0.406	0	0.406	0
256	0.410	0	0.409	0	0.410	0

Hence, we only need to simplify $H(S|\widetilde{S}, \widetilde{S}_d, \widetilde{S}_a, W)$. Observe that

$$H(S|\widetilde{S}, \widetilde{S}_d, \widetilde{S}_a, W) = H(S|\widetilde{S}, \widetilde{S}_a = E, W)p_a = H(S|\widetilde{S})p_a \tag{4.14}$$

and therefore

$$I(S; \widetilde{S}|\widetilde{S}_d, \widetilde{S}_a, W) = I(S; \widetilde{S}|W)p_d \ . \tag{4.15}$$

■

Evaluation of Theorem 4.1 and Theorem 4.2 leads to the achievability results presented in Tables 4.4.1 and 4.4.2. We assumed that the analog attacker has to destroy twice as much of the foil as the digital attacker has to. In Table 4.4.1 we took the conservative approach of assuming that the digital attacker destroys 10% of the PUF cells, i.e. $p_d = 0.1, p_a = 0.2$. Table 4.4.2 shows the results for $p_d = 0.18, p_a = 0.36$, which is the value corresponding to destroying one electrode in both layers of the foil as discussed in [IOK+18] considering a hole diameter $> 300\mu m$.

In both tables we considered various numbers of quantization levels and input quantizer strategies. From the proofs of Theorem 4.1 and Theorem 4.2 it is easy to see that choosing a suboptimal input quantizer still gives an achievable lower bound on the secret key rate.

In this work, we consider equidistant and equiprobable input quantization. Furthermore, we use an optimization algorithm aiming to find the input quantizer maximizing the achievable secret key rate according to Theorem 4.1 and Theorem 4.2.

The results show that in case we have to cope with the digital attacker that the equiprobable input quantization performs better than equidistant quantization. For the analog attacker the opposite is the case as we increase the number of quantization levels. For a small number of quantization levels equiprobable quantization still

Table 4.4.2: Achievable asymptotic rates for the PUF-channel for weak and strong attackers, different input quantization alphabets and quantization strategies, erasure probability $p_d = 0.18$ for the digital attacker and $p_a = 0.36$ for the analog attacker, $\sigma_P = 2241, \sigma_N = 129$

quantizer levels	equidist. digital	equidist. analog	equiprob. digital	equiprob. analog	optimized digital	optimized analog
2	0.18	0.18	0.18	0.18	0.18	0.18
4	0.211	0.211	0.36	0.36	0.36	0.36
8	0.353	0.353	0.536	0.524	0.537	0.533
16	0.523	0.523	0.640	0.356	0.654	0.600
32	0.677	0.591	0.687	0	0.688	0.613
64	0.700	0.061	0.716	0	0.716	0.061
128	0.727	0	0.731	0	0.731	0
256	0.738	0	0.737	0	0.738	0

performs better.

The reason for that behaviour is that the analog attacker obtains a perfect duplicate of the PUF cell measured during enrollment in case the respective cell is not erased by the holes he is required to drill to perform measurements. Especially if the measurement during the reconstruction by the legitimate user is unreliable, the analog measurements provide valuable additional information to the analog attacker that the legitimate user does not have. The input quantization has a substantial effect on the reliability of the measurement during reconstruction. Once an interval length at the output quantizer (influenced by the input quantizer) is below a certain threshold a measurement result of this value becomes unreliable. In this case the additional information provided by the analog measurement substantially increases due to the insecurity of the reconstruction value for the legitimate user.

This explains why equidistant input quantization performs better for a larger number of quantization levels if the analog attacker is considered. For a small amount of quantization levels the intervals are anyway large enough and the dominant factor is the entropy of the input, which is obviously maximized for the uniform input distribution achieved by the equiprobable input quantizer. For the digital attacker we do not have this trade-off and hence equiprobable quantization always performs better than equidistant quantization.

4.5 Secret Sharing using One-Way Communication for Finite Lengths

For this section we stick to the notation used in Section 2.2.2 that is commonly used in secret sharing with common randomness. In particular X, Y and Z are not related to the foil PUF in particular but rather form a general source of randomness via the pmf P_{XYZ}. After this section X shall again be considered the analog measurement during enrollment and Y shall again be considered the analog measurement during reconstruction.

Because of this inherent limitation of the construction used in [HTW16], we take the approach of utilizing the construction in the achievability proof sketch of Theorem 2.6 also in the finite blocklength case.

Theorem 4.3. *For the secret key rate with maximum secrecy δ and average error probability ε it holds that*

$$\widetilde{R}^*_{max}(n,\varepsilon,\delta) \geq \widetilde{C}_S - \sqrt{\frac{V_1}{n}}Q^{-1}(\varepsilon) - \sqrt{\frac{V_2}{n}}Q^{-1}(\delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \quad , \tag{4.16}$$

where

$$V_1 = \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}}} P_{XY}(x,y)\log_2^2\left(\frac{P_{XY}(x,y)}{\frac{1}{|\mathcal{X}|}P_Y(y)}\right) - D\left(P_{XY}||P_{\mathcal{X}}^{unif}P_Y\right)^2 \tag{4.17}$$

and

$$V_2 = \sum_{\substack{x\in\mathcal{X}\\ z\in\mathcal{Z}}} P_{XZ}(x,z)\log_2^2\left(\frac{P_{XZ}(x,z)}{\frac{1}{|\mathcal{X}|}P_Z(z)}\right) - D\left(P_{XZ}||P_{\mathcal{X}}^{unif}P_Z\right)^2 \tag{4.18}$$

with $P_{\mathcal{X}}^{unif}$ denoting the uniform distribution over the input alphabet $\mathcal{X}$. This rate can be achieved using one-way communication over the public channel.

Proof. As in the achievability proof sketch for Theorem 2.6, we use a secrecy codebook for the degraded wiretap channel sampled according to a distribution P_U, where in this particular case we define P_U to be uniform over the alphabet $\mathcal{X}$. We mask the codeword symbolwise by computing $u_i + x_i$ using the common randomness provided by the source P_{XYZ} and send the result over the public channel to Bob. Again the artificially created wiretap channel with input U and outputs $(U+X, Y)$ and $(U+X, Z)$ can be used to show achievability results for secret sharing using common randomness, this time in the finite blocklength regime. To compute the achievable secret key rates we apply Theorem 2.10. Notice that the secrecy definitions for (degraded) wiretap channels and secret sharing using common randomness match each other. Hence, the secrecy condition of the secrecy code implies the secrecy definition for secret sharing using common randomness.

For the first channel dispersion term V_1 the legitimate user channel from U to $(U+X,Y)$ is relevant. Hence, we have

$$V_1 = \sum_{u\in\mathcal{X}} P_U(u)\left(\sum_{\substack{\tilde{x}\in\mathcal{X}\\ y\in\mathcal{Y}}} P_{U\oplus X,Y|U}(\tilde{x},y|u)\log_2^2\left(\frac{P_{U\oplus X,Y|U}(\tilde{x},y|u)}{P_{U\oplus X,Y}(\tilde{x},y)}\right) - D\left(P_{U\oplus X,Y|U=u}||P_{U\oplus X,Y}\right)^2\right) . \tag{4.19}$$

Next we analyze the terms $P_{U\oplus X,Y|U}(\tilde{x},y|u)$ and $P_{U\oplus X,Y}(\tilde{x},y)$. We have

$$P_{U\oplus X,Y|U}(\tilde{x},y|u) = P_{U\oplus X|U}(\tilde{x}|u)P_{Y|U\oplus X,U}(y|\tilde{x},u) = P_X(\tilde{x}-u)P_{Y|X}(y|\tilde{x}-u) \tag{4.20}$$

because U is uniformly distributed over $\mathcal{X}$ and independent of X,Y.

Furthermore, it holds that

$$P_{U\oplus X,Y}(\tilde{x},y) = P_{U\oplus X}(\tilde{x})P_{Y|U\oplus X}(y|\tilde{x}) = \frac{1}{|\mathcal{X}|}P_Y(y) \tag{4.21}$$

again because U is uniformly distributed over $\mathcal{X}$ and independent of X,Y. Notice that $U\oplus X$ is independent of Y even though X and Y may not be independent. This is basically because X is encrypted by a one-time pad using U as its key.

Equation (4.17) follows because the sum in the bracket goes over the entire alphabet sets $\mathcal{X}$ and $\mathcal{Y}$. Notice that the sums do not depend on the choice of U in the outer sum. The same holds for the sums defining the divergence term.

An analogous argument holds for V_2 in Equation (4.18). ■

Theorem 4.4. *Let us assume that we use the same communication protocol as in the proof of Theorem 4.3. In this case for the secret key rate with average secrecy δ and average error proabability ε it holds that*

$$\widetilde{R}^*_{avg} \leq \widetilde{C}_S - \sqrt{\frac{V_c}{n}}Q^{-1}(\varepsilon+\delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) , \tag{4.22}$$

where

$$V_c = \sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}\\ z\in\mathcal{Z}}} P_{XYZ}(x,y,z)\log_2^2\left(\frac{P_{XYZ}(x,y,z)}{P_{XZ}(x,z)P_{Y|Z}(y|z)}\right) - D(P_{XYZ}||P_{Y|Z}P_{XZ})^2 . \tag{4.23}$$

This upper bound holds in particular for the secret sharing problem using only one-way communication over the public channel.

Proof. As mentioned in the theorem our goal is again to use a wiretap code for the

channel with uniformly sampled input U and with outputs $(U+X, Y)$ and $(U+X, Z)$. Hence, an upper bound on the secrecy rate for a wiretap code of this channel is also an upper bound on the achievable secret key rate of the communication protocol for the secret sharing problem.

Using the upper bound for wiretap codes in Theorem 2.10 it holds that

$$V_c = \sum_{u\in\mathcal{X}} P_U(u) \Bigg(\sum_{\substack{\tilde{x}\in\mathcal{X}\\ y\in\mathcal{Y}\\ z\in\mathcal{Z}}} P_{U\oplus X,Y,Z|U}(\tilde{x},y,z|u) \log_2^2 \left(\frac{P_{U\oplus X,Y,Z|U}(\tilde{x},y,z|u)}{P_{U\oplus X,Z|U}(\tilde{x},z|u) P_{U\oplus X,Y|U\oplus X,Z}(\tilde{x},y|\tilde{x},z)} \right) - D(P_{U\oplus X,Y,Z|U=u} || P_{U\oplus X,Y|U\oplus X,Z} P_{U\oplus X,Z|U=u})^2 \Bigg) \ . \tag{4.24}$$

To simplify Equation (4.24) we observe

$$\begin{aligned} P_{U\oplus X,Y,Z|U}(\tilde{x},y,z|u) &= P_{U\oplus X|U}(\tilde{x}|u) P_{Y|U\oplus X,U}(y|\tilde{x},u) P_{Z|Y,U\oplus X,U}(z|y,\tilde{x},u) \\ &= P_X(\tilde{x}-u) P_{Y|X}(y|\tilde{x}-u) P_{Z|Y,X}(z|y,\tilde{x}-u) \ , \end{aligned} \tag{4.25}$$

$$P_{U\oplus X,Z|U}(\tilde{x},z|u) = P_{U\oplus X|U}(\tilde{x}|u) P_{Z|U\oplus X,U}(z|\tilde{x},u) = P_X(\tilde{x}-u) P_{Z|X}(z|\tilde{x}-u) \tag{4.26}$$

and

$$P_{U\oplus X,Y|U\oplus X,Z}(\tilde{x},y|\tilde{x},z) = P_{Y|Z}(y|z) \ . \tag{4.27}$$

The basic idea to show all of these simplifications is to use the facts that U is uniformly distributed over $\mathcal{X}$ and independent of X, Y, Z and that $U \oplus X$ is independent of Y and Z because X is encrypted by a one-time pad using the uniformly distributed U as a key.

As in the proof of Theorem 4.3 the inner sum in Equation (4.24) goes over the entire alphabet sets $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$. Hence, the sums do not depend on the specific value of U since it only leads to an index shift within the sum. The same argument holds for the divergence term. ■

Corollary 4.1. *Let us consider a source of common randomness following a joint pmf P_{XYZ} for which it holds that $X \multimap Y \multimap Z$. Then it holds that the upper bound on the maximal secrecy rate for wiretap codes in Theorem 2.10 is not tight for the degraded wiretap channel formed by the uniformly sampled input U (over $\mathcal{X}$), legitimate user output $(U \oplus X, Y)$ and eavesdropper output $(U \oplus X, Z)$.*

Proof. As already emphasized in Remark 2.6 the communication protocol over the public channel achieving the secret key rate in Theorem 2.11 requires two-way communication over the public channel while the construction of the upper bound in Theorem 4.4 only requires one-way communication. Hence, in order to prove the corollary, it suffices to show that the upper bound in Theorem 4.4 exceeds the maximal secret key rate in Theorem 2.11. We will show in the following that this indeed holds.

The difference between the rates in Theorems 4.4 and 2.11 lies in the channel dispersion terms V_c and V_c', respectively.

Notice that the sums involving the $\log_2^2$ terms are equivalent. Hence, we focus on the divergence terms in both equations. We have

$$\begin{aligned} D(P_{XYZ}||P_{Y|Z}P_{XZ})^2 &= \left(\sum_{\substack{x\in\mathcal{X}\\ y\in\mathcal{Y}\\ z\in\mathcal{Z}}} P_{XYZ}(x,y,z) \log_2 \left(\frac{P_{XYZ}(x,y,z)}{P_{Y|Z}(y|z)P_{XZ}(x,z)} \right) \right)^2 \\ &= \left(\sum_{x\in\mathcal{X}} P_X(x) \sum_{\substack{y\in\mathcal{Y}\\ z\in\mathcal{Z}}} P_{YZ|X}(y,z|x) \log_2 \left(\frac{P_{YZ|X}(y,z|x)}{P_{Y|Z}(y|z)P_{Z|X}(z|x)} \right) \right)^2 \end{aligned} \tag{4.28}$$

and

$$\begin{aligned} &\sum_{x\in\mathcal{X}} P_X(x) D(P_{YZ|X=x}||P_{Y|Z}P_{Z|X=x})^2 \\ &= \sum_{x\in\mathcal{X}} P_X(x) \left(\sum_{\substack{y\in\mathcal{Y}\\ z\in\mathcal{Z}}} P_{YZ|X}(y,z|x) \log_2 \left(\frac{P_{YZ|X}(y,z|x)}{P_{Y|Z}(y|z)P_{Z|X}(z|x)} \right) \right)^2 . \end{aligned} \tag{4.29}$$

Because $f(x) = x^2$ is a strictly convex function we have that

$$D(P_{XYZ}||P_{Y|Z}P_{XZ})^2 > \sum_{x\in\mathcal{X}} P_X(x) D(P_{YZ|X=x}||P_{Y|Z}P_{Z|X=x})^2 \tag{4.30}$$

and hence the statement follows. ■

Remark 4.1. *Corollary 4.1 shows for sources of common randomness P_{XYZ} for which $X \multimap Y \multimap Z$ that the upper bound obtained in Theorem 4.4 is strictly larger than the rate obtained in Theorem 2.11. The secret key rate of Theorem 2.11 also provides an upper bound on the secret key rate for protocols with one-way communication. Notice though that Theorem 2.11 requires the random variables X, Y, Z to be connected via a Markov chain while for Theorem 4.4 can also be applied if this is not the case.*

Equipped with the knowledge of this section we next tackle the problem of securing the HDA against the attack scenarios mentioned in Section 4.3.1 for finite lengths.

4.6 Achievability and Converse Bounds for Finite Lengths

The results presented in the previous section are for the asymptotic setting in the sense that they hold if the number of capacitive cells in the PUF goes to infinity and the fraction of the erasures for the attacker are determined by the erasure channel parameters p_d and p_a for the digital and analog attacker model. Of course as a first order approximation one could compute the key capacity C_{key}^q and estimate the dimension of the secrecy code for the resulting (degraded) wiretap channel by $C_{key}^q n$. As for the channel models discussed in Section 2.3 to estimate the code dimension in that way is highly inaccurate if the blocklength is of small to moderate size. In general the achievable code dimension is of significantly smaller size. We note that the erasure channel assumption is only an approximation of the reality since for our attacker model the number of erasures is assumed to be constant and equal to $p_d n$ or $p_a n$ for the digital and the analog attacker model, respectively. Nevertheless, we consider the results obtaining from this to be rather accurate because these values are equal to the expectation for the erasure channels. Notice here that erasures in the digital attacker model imply erasures in the analog attacker model. The erasures for the analog attacker are then added on top.

Using Theorem 4.3 to estimate the achievable code rate leads to a much more precise estimate for small to moderate blocklengths. Similarly, applying Theorem 4.4 is much more precise to estimate a converse result on the secret key rate for the HDA. However, using Theorem 2.11 enables us to find an even tighter converse in case we only need to protect the design against the digital attacker.

The following observation is helpful and used implicitly throughout this section. Empirically, we found that $I(S;\widetilde{S}|W) \approx I(S;\widetilde{S})$. Basically we observed no difference between these quantities even though we were not able to formally prove this statement. Notice that this does not mean that the helper data has not been utilized on the right hand side of the equation. Rather it means that the channel matrix on the right hand side is constructed by averaging over all possible values of W rather than constructing the channel matrix for each realization w, computing $I(S;\widetilde{S}|W=w)$ and then averaging with respect to $f_W(w)$. This effect is desirable as it shows that it is sufficient to construct a single codebook independent of the helper data W, rather than constructing different codebooks for different values of W. This behaviour makes perfect sense considering that $\widetilde{S}$ is an approximation of S and W is independent of S due to the zero leakage condition (see Section 2.7.6).

The potential inaccuracy by assuming $I(S;\widetilde{S}) = I(S;\widetilde{S}|W)$ only induces a rate penalty for the scheme. In terms of security this is not a problem as W is still perfectly utilized and the codebook is chosen by the legitimate users. Furthermore, it makes the application of Theorem 2.10 significantly easier.

In the following, we are investigating which finite rates can be achieved by HDAs

with different parameters for both attacker models. We are in particular interested in this rate as the code dimension in the HDA has to be at least as large as the targeted security level. A lower bound on the code rate (which also depends on the blocklength for finite n) leads to an upper bound on the required amount of capacitive cells while a lower bound can be used to prove impossibility results, i.e. it enables to show that a certain amount of cells is absolutely required for target values in terms of reliability and security of the PUF.

As in Section 4.4 we first consider the HDA scheme only covering the digital attacker.

4.6.1 Digital Attacker

Theorem 4.5. *The maximal achievable rate $\widetilde{R}_{max}^{key,dig}(n,\varepsilon,\delta)$ for the HDA achieving a maximum secrecy level δ against the digital attacker described in Section 4.3.1 with error probability ε during regular device operation is lower bounded by*

$$\widetilde{R}_{max}^{key,dig}(n,\varepsilon,\delta) \geq R_{asymp,dig}^{q} - \sqrt{\frac{V_1}{n}}Q^{-1}(\varepsilon) - \sqrt{\frac{V_2}{n}}Q^{-1}(\delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \ , \tag{4.31}$$

where

$$R_{asymp,dig}^{q} := I(S;\widetilde{S})p_d \ , \tag{4.32}$$

$$V_1 = \sum_{\substack{s\in\mathcal{S}\\ \tilde{s}\in\mathcal{S}}} P_{S,\widetilde{S}}(s,\tilde{s}) \log_2^2\left(\frac{P_{S,\widetilde{S}}(s,\tilde{s})}{\frac{1}{|\mathcal{S}|}P_{\widetilde{S}}(\tilde{s})}\right) - D(P_{S,\widetilde{S}}||P_{\mathcal{S}}^{unif}P_{\widetilde{S}})^2 \tag{4.33}$$

and

$$\begin{aligned} V_2 = &\sum_{\substack{s\in\mathcal{S}\\ \tilde{s}\in\mathcal{S}}} P_{S,\widetilde{S}}(s,\tilde{s})(1-p_d)\log_2^2\left(\frac{P_{S,\widetilde{S}}(s,\tilde{s})}{\frac{1}{|\mathcal{S}|}P_{\widetilde{S}}(\tilde{s})}\right) + \sum_{s\in\mathcal{S}} P_S(s)p_d\log_2^2\left(\frac{P_S(s)}{\frac{1}{|\mathcal{S}|}}\right) \\ &- \left[(1-p_d)D\left(P_{S,\widetilde{S}}||P_{\mathcal{S}}^{unif}P_{\widetilde{S}}\right) + p_d D\left(P_S||P_{\mathcal{S}}^{unif}\right)\right]^2 \ . \end{aligned} \tag{4.34}$$

Proof. The statement follows from Theorem 4.3 by setting $S \equiv X$, $\widetilde{S} \equiv Y$ and $\widetilde{S}_d \equiv Z$. V_1 is directly obtained by plugging in the respective random variables. For the computation of V_2 we note that the probability that $P_{\widetilde{S}_d|S}(E|s) = p_d$ irrespective of s. In case no erasure occurs it holds that $P_{\widetilde{S}_d|S}(\tilde{s}|s) = P_{\widetilde{S}|S}(\tilde{s}|s)(1-p_d)$. The rest follows easily from the definition of V_2 in Theorem 4.3. ■

Theorem 4.6. *The secret key rate for a source of common randomness with distribution $P_{S,\widetilde{S},\widetilde{S}_d}$ is*

$$\widetilde{R}_{avg}^{*}(n,\varepsilon,\delta) = C_S - \sqrt{\frac{V_c'}{n}}Q^{-1}(\varepsilon+\delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \ , \tag{4.35}$$

where

$$V_c' = \sum_{s\in\mathcal{S}} P_S(s) \sum_{\tilde{s}\in\mathcal{S}} P_{\widetilde{S}|S}(\tilde{s}|s) p_d \log_2^2\left(\frac{P_{\widetilde{S}|S}(\tilde{s}|s)}{P_{\widetilde{S}}(\tilde{s})}\right) - \sum_{s\in\mathcal{S}} P_S(s) p_d^2 D\left(P_{\widetilde{S}|S=s}||P_{\widetilde{S}}\right)^2 \quad . \tag{4.36}$$

Furthermore, it holds that

$$\widetilde{R}^*_{avg}(n,\varepsilon,\delta) \geq \widetilde{R}^{key}_{avg}(n,\varepsilon,\delta) \ , \tag{4.37}$$

where $\widetilde{R}^q_{avg,dig}(n,\varepsilon,\delta)$ denotes the maximal achievable rate for the HDA achieving an average security level δ against the digital attacker

Proof. The statement follows from Theorem 2.11 by setting $S \equiv X$, $\widetilde{S} \equiv Y$ and $\widetilde{S}_d \equiv Z$. Notice that $S, \widetilde{S}, \widetilde{S}_d$ form the Markov chain $S \multimap\!\circ\!\!- \widetilde{S} \multimap\!\circ\!\!- \widetilde{S}_d$ which is necessary for Theorem 2.11 to be applicable.

By the definition of $\widetilde{S}_d$ we have that $\widetilde{S}_d$ is either equal to $\widetilde{S}$ or the erasure event E. We have that

$$\begin{aligned} P_{S,\widetilde{S},\widetilde{S}_d}(s,\tilde{s},\tilde{s}) &= P_S(s) P_{\widetilde{S}|S}(\tilde{s}|s)(1-p_d) \\ P_{S,\widetilde{S},\widetilde{S}_d}(s,\tilde{s},E) &= P_S(s) P_{\widetilde{S}|S}(\tilde{s}|s) p_d. \end{aligned} \tag{4.38}$$

Furthermore, it holds that $P_{\widetilde{S}|\widetilde{S}_d}(\tilde{s}|\tilde{s}) = 1$ and $P_{\widetilde{S}|\widetilde{S}_d}(\tilde{s}|E) = P_{\widetilde{S}}(\tilde{s})$.

Using these statements and applying them to Theorem 2.11 we observe that all terms for which $\widetilde{S}_d \neq E$ go to zero. Equation (4.36) follows by considering the remaining terms for which $\widetilde{S}_d = E$. ■

By Remark 4.1 we know that the upper bound given in Theorem 4.6 is tighter than applying Theorem 4.4. Using Theorem 4.5 and Theorem 4.6 we obtain the achievability and converse results presented in Tables 4.6.1, 4.6.2, 4.6.3 and 4.6.4.

The results show that different amounts of input quantization intervals can significantly change the achievable rates for the foil PUF. Notice that for the achievability bounds, increasing the amount of quantization intervals does not necessarily decrease the required number of PUF cells. 2 Bit quantization is optimal from that perspective. The converse bounds decrease with an increasing the amount of quantization intervals though. From the tables we observe that an increased demand in PUF reliability only has a small influence on the required number of cells, whereas the erasure probability has a much larger impact. We also observe that higher security levels require more PUF cells as was to be expected.

4.6.2 Analog Attacker

In this section we basically perform a similar analysis for the analog attacker as we did for the digital attacker.

Table 4.6.1: Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p_d = 0.1$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization, $\sigma_P = 2241, \sigma_N = 129$

quantizer	ach.128b	conv.128b	ach.192b	conv.192b	ach.256b	conv.256b
2	3645	1902	5499	2655	7354	3391
4	2664	1117	4025	1516	5386	1902
8	3178	850	4635	1128	6072	1398
16	5502	779	7768	1019	9977	1250
32	5390	744	7609	970	9773	1187
64	5509	726	7768	943	9968	1152

Table 4.6.2: Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p_d = 0.18$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization, $\sigma_P = 2241, \sigma_N = 129$

quantizer	ach.128b	conv.128b	ach.192b	conv.192b	ach.256b	conv.256b
2	1938	1038	2923	1454	3909	1860
4	1399	606	2113	825	2828	1038
8	1508	459	2216	612	2916	760
16	2194	420	3128	552	4042	680
32	2150	401	3064	525	3959	644
64	2179	390	3102	510	4004	625

Theorem 4.7. *The maximal achievable rate $\widetilde{R}_{max}^{key,ana}(n,\varepsilon,\delta)$ for the HDA achieving a maximum secrecy level δ against the analog attacker described in Section 4.3.1 with error probability ε during regular device operation is lower bounded by*

$$\widetilde{R}_{max}^{key,ana}(n,\varepsilon,\delta) \geq R_{asymp,ana}^{q,lower} - \sqrt{\frac{V_1}{n}}Q^{-1}(\varepsilon) - \sqrt{\frac{V_2}{n}}Q^{-1}(\delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \ , \quad (4.39)$$

where

$$R_{asymp,ana}^{q,lower} := I(S;\widetilde{S}|W)(1-p_a+p_d) - H(S)(1-p_a) \ , \quad (4.40)$$

$$V_1 = \sum_{\substack{s\in\mathcal{S}\\ \tilde{s}\in\mathcal{S}}} P_{S,\widetilde{S}}(s,\tilde{s}) \log_2^2\left(\frac{P_{S,\widetilde{S}}(s,\tilde{s})}{\frac{1}{|\mathcal{S}|}P_{\widetilde{S}}(\tilde{s})}\right) - D(P_{S,\widetilde{S}}||P_{\mathcal{S}}^{unif}P_{\widetilde{S}})^2 \quad (4.41)$$

Table 4.6.3: Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p = 0.1$, PUF reliability $\varepsilon = 10^{-9}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization

quantizer	ach.128b	conv.128b	ach.192b	conv.192b	ach.256b	conv.256b
2	3645	2106	5499	2887	7354	3647
4	2665	1286	4026	1703	5388	2106
8	3345	1006	4834	1300	6299	1582
16	6050	940	8414	1194	10705	1437
32	5927	903	8243	1142	10488	1370
64	6068	884	8427	1114	10712	1335

Table 4.6.4: Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p = 0.18$, PUF reliability $\varepsilon = 10^{-9}$ and security levels 128, 192 and 256 bit, digital attacker, equiprobable input quantization

quantizer	ach.128b	conv.128b	ach.192b	conv.192b	ach.256b	conv.256b
2	1938	1145	2923	1575	3909	1994
4	1400	693	2114	923	2828	1145
8	1571	539	2291	701	3002	856
16	2382	504	3350	643	4293	777
32	2334	483	3282	614	4205	740
64	2370	472	3327	599	4259	720

and

$$V_2 = p_d \sum_{s\in\mathcal{S}} P_S(s) \log_2^2\left(\frac{P_S(s)}{\frac{1}{|\mathcal{S}|}}\right) + (p_a - p_d) \sum_{\substack{s\in\mathcal{S}\\ \tilde{s}\in\mathcal{S}}} P_{S,\widetilde{S}}(s,\tilde{s}) \log_2^2\left(\frac{P_S(s) P_{\widetilde{S}|S}(\tilde{s}|s)}{\frac{1}{|\mathcal{S}|} P_{\widetilde{S}}(\tilde{s})}\right)$$
$$+ (1-p_a)\log_2^2(|\mathcal{S}|) - \left[\log_2(|\mathcal{S}|) + I(S;\widetilde{S})(p_a - p_d) - p_a H(S)\right]^2 \tag{4.42}$$

Proof. To prove the statement we use Theorem 4.3 and set $S \equiv X$, $\widetilde{S} \equiv Y$ and $(\widetilde{S}_d, \widetilde{S}_a) \equiv Z$. The channel for the legitimate user does not change compared to the digital attacker. Hence, the dispersion term V_1 does not change compared to Theorem 4.5.

Therefore, we only need to provide a proof for V_2. For $\widetilde{S}_d \neq E$ and $\widetilde{S}_a \neq E$, we have

that

$$\begin{aligned} P_{S,\widetilde{S}_d,\widetilde{S}_a}(s,\tilde{s},s) &= P_{S,\widetilde{S}}(s,\tilde{s})(1-p_a) \\ P_{\widetilde{S}_d,\widetilde{S}_a}(\tilde{s},s) &= P_S(s)P_{\widetilde{S}|S}(\tilde{s}|s)(1-p_a) \ . \end{aligned} \tag{4.43}$$

Furthermore, for $\widetilde{S}_d = E$ and $\widetilde{S}_a = E$

$$\begin{aligned} P_{S,\widetilde{S}_d,\widetilde{S}_a}(s,E,E) &= P_S(s)p_d \\ P_{\widetilde{S}_d,\widetilde{S}_a}(E,E) &= p_d \end{aligned} \tag{4.44}$$

and for $\widetilde{S}_d \neq E$ and $\widetilde{S}_a = E$

$$\begin{aligned} P_{S,\widetilde{S}_d,\widetilde{S}_a}(s,\tilde{s},E) &= P_S(s)P_{\widetilde{S}|S}(\tilde{s}|s)(p_a-p_d) \\ P_{\widetilde{S}_d,\widetilde{S}_a}(\tilde{s},E) &= P_{\widetilde{S}}(\tilde{s})(p_a-p_d) \end{aligned} \tag{4.45}$$

Using these identities and Theorem 4.3 we obtain

$$\begin{aligned} V_2 =& p_d \sum_{s\in\mathcal{S}} P_S(s)\log_2^2\left(\frac{P_S(s)}{\frac{1}{|\mathcal{S}|}}\right) + (p_a-p_d)\sum_{\substack{s\in\mathcal{S}\\ \tilde{s}\in\mathcal{S}}} P_{S,\widetilde{S}}(s,\tilde{s})\log_2^2\left(\frac{P_S(s)P_{\widetilde{S}|S}(\tilde{s}|s)}{\frac{1}{|\mathcal{S}|}P_{\widetilde{S}}(\tilde{s})}\right) \\ &+ (1-p_a)\log_2^2(|\mathcal{S}|) - \Big[\log_2(|\mathcal{S}|) + I(S;\widetilde{S})(p_a-p_d) - p_aH(S)\Big]^2 \end{aligned} \tag{4.46}$$

after some elementary algebraic steps. ■

Theorem 4.8. *The maximal achievable rate $\widetilde{R}_{avg}^{key,ana}(n,\varepsilon,\delta)$ for the HDA achieving a average secrecy level δ against the analog attacker described in Section 4.3.1 with error probability ε during regular device operation is upper bounded by*

$$\widetilde{R}_{avg}^{key,ana}(n,\varepsilon,\delta) \leq R_{asymp,ana}^{q,upper} - \sqrt{\frac{V_c}{n}}Q^{-1}(\varepsilon+\delta) + \mathcal{O}\left(\frac{\log(n)}{n}\right) \tag{4.47}$$

with

$$R_{asymp,ana}^{q,upper} := I(S;\widetilde{S}|W)p_d \tag{4.48}$$

and

$$V_c = p_d \sum_{\substack{s\in\mathcal{S}\\ \tilde{s}\in\mathcal{S}}} P_{S,\widetilde{S}}(s,\tilde{s})\log_2^2\left(\frac{P_{\widetilde{S}|S}(\tilde{s}|s)}{P_{\widetilde{S}}(\tilde{s})}\right) - p_d^2 I(S;\widetilde{S})^2 \ . \tag{4.49}$$

Proof. To prove the statement we use Theorem 4.4 and set $S \equiv X$, $\widetilde{S} \equiv Y$ and

$(\widetilde{S}_d, \widetilde{S}_a) \equiv Z$. For $\widetilde{S}_d \neq E$ and $\widetilde{S}_a \neq E$ we have that

$$\begin{aligned} P_{S,\widetilde{S},\widetilde{S}_d,\widetilde{S}_a}(s,\tilde{s},\tilde{s},s) &= P_{S,\widetilde{S}}(s,\tilde{s})(1-p_a) \\ P_{S,\widetilde{S}_d,\widetilde{S}_a}(s,\tilde{s},s) &= P_S(s)P_{\widetilde{S}|S}(\tilde{s}|s)(1-p_a) \\ P_{\widetilde{S}|\widetilde{S}_d,\widetilde{S}_a}(\tilde{s}|\tilde{s},s) &= 1 \ , \end{aligned} \tag{4.50}$$

for $\widetilde{S}_d = E$ and $\widetilde{S}_a = E$ we have that

$$\begin{aligned} P_{S,\widetilde{S},\widetilde{S}_d,\widetilde{S}_a}(s,\tilde{s},E,E) &= P_{S,\widetilde{S}}(s,\tilde{s})p_d \\ P_{S,\widetilde{S}_d,\widetilde{S}_a}(s,E,E) &= P_S(s)p_d \\ P_{\widetilde{S}|\widetilde{S}_d,\widetilde{S}_a}(\tilde{s}|E,E) &= P_{\widetilde{S}}(\tilde{s}) \end{aligned} \tag{4.51}$$

and for $\widetilde{S}_d \neq E$ and $\widetilde{S}_a = E$ it holds that

$$\begin{aligned} P_{S,\widetilde{S},\widetilde{S}_d,\widetilde{S}_a}(s,\tilde{s},\tilde{s},E) &= P_S(s)P_{\widetilde{S}|S}(\tilde{s}|s)(p_a-p_d) \\ P_{S,\widetilde{S}_d,\widetilde{S}_a}(s,\tilde{s},s) &= P_S(s)p_d \\ P_{\widetilde{S}|\widetilde{S}_d,\widetilde{S}_a}(\tilde{s}|\tilde{s},s) &= P_{\widetilde{S}}(\tilde{s}) \end{aligned} \tag{4.52}$$

Using Theorem 4.4 and the identities outlined above directly gives the desired result after some elementary steps. ■

Remark 4.2. *Notice that it is to be assumed that the converse bound is of poor quality as the analog erasure probability p_a has no influence on the bound. The approach taken for the digital attacker to use Theorem 2.11 to find an upper bound on the maximal rate of the HDA cannot be done for the analog attacker in a straightforward manner. The reason is that Theorem 2.11 is only applicable if S, $\widetilde{S}$ and $(\widetilde{S}_d, \widetilde{S}_a)$ form a Markov chain, which is not the case.*

The results for applying Theorem 4.7 and Theorem 4.8 to the HDA in the analog attacker scenario are presented in Table 4.6.5 and Table 4.6.6. Achievability as well as converse results on the required number of PUF cells are given. Comparing the achievability results to the results for the digital attacker, we observe that for a small amount of quantization levels are almost identical. Even though we cannot say the same for the converse results this is an interesting observation. It suggests that potentially it is better to use coarse input quantizers. Aside from requiring less PUF cells from an achievability bound perspective quantizers with fewer levels are cheaper and easier to build. Also notice that the output quantizer levels need to be adjusted according to the helper data. In this respect this observation becomes even more important than in the enrollment phase.

Table 4.6.5: Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p_d = 0.18$ and $p_a = 0.36$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, analog attacker, equiprobable quantization

quantizer	ach.128b	conv.128b	ach.192b	conv.192b	ach.256b	conv.256b
2	1938	902	2923	1295	3909	1683
4	1399	417	2113	607	2828	795
8	1511	239	2219	358	2918	478
16	5983	201	8470	301	10897	401
32	-	187	-	280	-	373
64	-	179	-	269	-	358

Table 4.6.6: Achievability (ach.) and converse (conv.) results on the number of necessary capacitive PUF cells for $p = 0.18$ and $p_a = 0.36$, PUF reliability $\varepsilon = 10^{-6}$ and security levels 128, 192 and 256 bit, analog attacker, equidistant quantization

quantizer	ach.128b	conv.128b	ach.192b	conv.192b	ach.256b	conv.256b
2	1938	902	2923	1295	3909	1683
4	2305	740	3482	1070	4659	1396
8	1679	363	2537	545	3397	726
16	1355	245	2044	367	2734	490
32	2184	190	3141	284	4081	379
64	-	183	-	275	-	366

4.7 Applying Converse Results in Security Analysis

In previous wiretap work [GXKF22b], the complexity of the attacker has been estimated by $H_{att} = -\sum_i^{n_s} p_{s,i} \log_2(p_{s,i})$, where n_s is the dimension of the code and $p_{s,i}$ denotes the symbol error rate of an attack on the information symbols of the polar code. For the parameters in the paper this leads to a claimed attack complexity of 100 bit for a legitimate user reliability $\varepsilon = 10^{-6}$.

However, according to the converse bounds outlined in this work (Theorem 4.6), it is impossible to reach the claimed security level of 100 bit for the specified reliability level $\varepsilon = 10^{-6}$. More specifically, the bound shows that at least 389 PUF cells with 8 quantization levels are required (neglecting the $\mathcal{O}$-error term in Theorem 4.6) rather than the proposed 128 PUF cells. Hence, assuming that the error term can be neglected, computation of the converse bound shows that the security analysis made in [GXKF22b] is flawed and the proposed construction does not reach the desired security level without leaking information about the secret through the helper data. This shows that our rather theoretical framework has substantial practical consequences.

4.8 Summary and Open Problems

Over the last years, several practical papers for key generation for PUF-based tamper protection were introduced. In this chapter, we analyzed the problem with information theoretical tools for additional insights and contribute to understanding and quantifying theoretical and practical limits.

We have given achievable rates under the constraint that the quantization helper data leaks no information about the quantized PUF response S^n obtained during the enrollment phase both in the asymptotic as well as the non-asymptotic setting for two different attacker models. Asymptotically we observed that for the digital attacker equiprobable input quantization performs better than equidistant input quantization in terms of achievable code rates. In case the analog attacker of concern to designer equiprobable quantization can be beneficial for larger quantization alphabets. The same observation holds in the finite length regime. Our results have practical merit since they show that coarse quantization is not only easier to implement but also often requires less capacitive cells on the foil. Furthermore, we presented converse bounds that showed that existing implementations cannot achieve a security level of 100 bits if leakage of information via the public quantization helper data is prevented. The achievability results allow to state an upper bound on the required number of capacitive PUF cells. Thereby, they permit to set design goals for PUFs with respect to the required security and reliability levels.

Our results show that we can achieve the number of required capacitive cells with an optimized implementation, e.g. through a feasible feature shrink. Consequently, by quantifying the fundamental limits of this PUF architecture this the results presented in this chapter contribute in guiding practical work towards theoretically secure and certifiable future implementations.

In this chapter we have not addressed the construction of a wiretap code actually achieving the desired security level against the proposed attackers. Standard constructions using polar codes are unlikely to work because the attacker's success rates cannot be determined via Monte-Carlo simulations. In the opinion of the author algebraic constructions are more promising because it is usually easier to calculate performance guarantees of the coding schemes. We consider this an interesting topic for further research.

Part II

Coding with Feedback for Selected Channels

5
Preliminaries

5.1 Notation

This section formally defines notations that are used throughout this part of the thesis. The set of integers $\{1, \ldots, n\}$ is denoted by $[n]$. The set of integers $\{i+1, i+2, \ldots, j\}$ is denoted by $(i, j]$. We denote sets by calligraphic letters, e.g. a set $\mathcal{A}$. For instance we can specify the set $\mathcal{A}$ to be binary by setting $\mathcal{A} := \{0, 1\}$ and to be q-ary by defining $\mathcal{A} = \mathcal{Q} := \{0, 1, \ldots, q-1\}$. A string of length n is denoted by $\boldsymbol{x} \in \mathcal{A}^n$, i.e. $\boldsymbol{x} = (x_1, \ldots, x_n)$. We use $x_i \in \mathcal{A}$ to denote the i-th element of the string $\boldsymbol{x}$ where $i \in [n]$. The set $\mathcal{A}^*$ contains all strings over the alphabet $\mathcal{A}$ of arbitrary length including the empty string which is denoted as ( ). We refer to the length of $\boldsymbol{x}$ as $|\boldsymbol{x}|$. Given two strings $\boldsymbol{x_1} \in \mathcal{A}^{n_1}$ and $\boldsymbol{x_2} \in \mathcal{A}^{n_2}$, we denote by $\boldsymbol{z} = (\boldsymbol{x}||\boldsymbol{y})$ the concatenation of $\boldsymbol{x}$ and $\boldsymbol{y}$, hence $\boldsymbol{z} \in \mathcal{A}^{n_1+n_2}$. It is also possible to write a string $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathcal{A}^n$ as a concatenation of its components, i.e. $\boldsymbol{x} = (x_1|| \ldots ||x_n)$. However, for n strings $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \in \mathcal{A}^*$, we distinguish the concatenation of strings $\boldsymbol{y} = (\boldsymbol{y}_1|| \ldots ||\boldsymbol{y}_n) \in \mathcal{A}^*$ from a tuple $\hat{\boldsymbol{y}}^{(n)} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \in (\mathcal{A}^*)^n$ containing several strings. The tuple $\hat{\boldsymbol{y}}^{(n)}$ can be uniquely split back into $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$, whereas for the concatenated string $\boldsymbol{y}$ this is not possible. Finally, we define the binary entropy function by $h(x) := -x \log_2 x - (1-x) \log_2 (1-x)$.

5.2 Distance Metrics

We use the following two distance notations. For any two strings $\boldsymbol{x} \in \mathcal{A}^*$ and $\boldsymbol{y} \in \mathcal{A}^*$ we denote as $\Delta(\boldsymbol{x}, \boldsymbol{y})$ the minimal number of deletions and insertions required to obtain $\boldsymbol{x}$ from $\boldsymbol{y}$. This quantity is frequently referred to as the *longest common subsequence* distance between strings $\boldsymbol{x}$ and $\boldsymbol{y}$ in the literature. Additionally, we use the notation $d_H(\boldsymbol{x}, \boldsymbol{y})$ to denote the Hamming distance between strings $\boldsymbol{x}$ and $\boldsymbol{y}$ of same length.

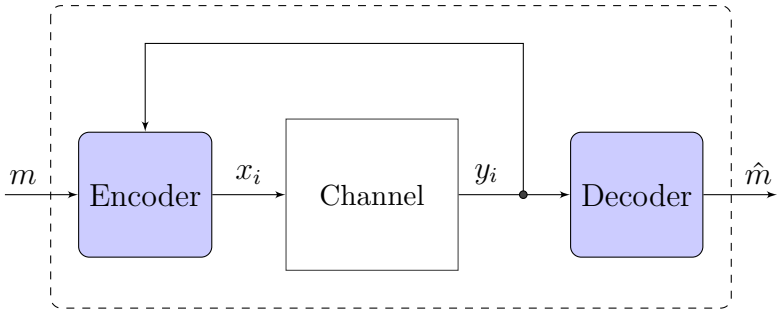

Figure 5.3.1: Channel with feedback

5.3 Coding with Feedback

A transmission scheme with feedback enables the sender to choose his encoding strategy adaptively according to previously received symbols within a transmission block at the receiver side. Throughout this work we assume the feedback to be noiseless and instantaneous, meaning that the encoder gets instant notification about the channel output and we specify the input alphabet of the channel by $\mathcal{X}$ and the output alphabet by $\mathcal{Y}$. The specification of the forward channel is equivalent to restricting which elements of the input alphabet $\mathcal{X}$ can be mapped to which elements of the output alphabet $\mathcal{Y}$.

We denote the set of possible messages by $\mathcal{M}$. The sender chooses a message contained in this set, say $m \in \mathcal{M}$, which he wants to send to the receiver. An encoding algorithm for a feedback channel of blocklength n is composed of a set of functions

$$c_i : \mathcal{M} \times \mathcal{Y}^{i-1} \to \mathcal{X}, \quad i \in \{1, \ldots, n\} \ . \tag{5.1}$$

The encoding algorithm is then constructed as

$$c(m, \mathbf{y}^{n-1}) = ((c_1(m), c_2(m, y_1), c_3(m, \mathbf{y}^2), \ldots, c_n(m, \mathbf{y}^{n-1})) \in \mathcal{X}^n \ , \tag{5.2}$$

where $\mathbf{y}^k := (y_1, \ldots, y_k)$ with $y_i \in \mathcal{Y}$ being the i^{th} received symbol. Moreover, the set of possible values for the received symbol y_i conditioned on c_i is defined by the channel. Suppose that at most t errors occur within a block of length n. For $m \in \mathcal{M}$, we define the set of output sequences for encoding strategy $c(m, \mathbf{y}^{n-1})$ by

$$\mathcal{Y}_t^n(m) := \left\{ \mathbf{y}^n \in \mathcal{Y}^n : \, d(\mathbf{y}^n, c(m, \mathbf{y}^{n-1})) \leq t \right\} \ , \tag{5.3}$$

where $d(\cdot, \cdot)$ denotes the distance metric we are considering for the respective channel. Within the definition in equation (5.3) it is implicit that the output sequences have to obey the specifications of the forward channel. As examples for a distance metric $d(\cdot, \cdot)$ we refer to the Hamming distance d_H or the longest common subsequence distance Δ

introduced in Section 5.2.

Definition 5.1. *An encoding strategy* (5.2) *is called* ***successful*** *if* $\mathcal{Y}_t^n(m_1) \cap \mathcal{Y}_t^n(m_2) = \emptyset$ *for all distinct messages* $m_1, m_2 \in \mathcal{M}$.

Remark 5.1. *An encoding strategy is guaranteed to cope with* t *errors inflicted by an adversary knowing the message* m *to be transmitted as well as encoding and decoding strategy of the communication system if and only if the encoding strategy is successful in the sense of Definition 5.1.*

Definition 5.2. *Let* $M_q(\Gamma, n, t)$ *denote the maximum number of messages in* $\mathcal{M}$ *for which there exists a successful encoding strategy for a channel* Γ *with input alphabet size* q *if the number of errors inflicted by the channel is upper bounded by* t. *A successful encoding strategy for* $M_q(\Gamma, n, t)$ *messages is said to be* ***optimal****.*

Definition 5.3. *We consider a channel* Γ. *Let the size of its input alphabet be* $q \geq 2$. *The capacity error function* $C_q^f(\Gamma, \tau)$ *of* Γ *with noiseless feedback denotes the supremum on the rates for which a successful algorithm exists depending on* $\tau = \frac{t}{n}$ *as the blocklength* n *goes to infinity with* t *denoting the maximum number of errors inflicted by the channel noise, i.e.,*

$$C_q^f(\Gamma, \tau) := \limsup_{n \to \infty} \frac{\log_q(M_q(\Gamma, n, \lceil \tau n \rceil))}{n} \quad . \tag{5.4}$$

The capacity error function for $\tau = 1$ *is denoted by* $C_{q,0}^f(\Gamma)$ *and has been introducted by Shannon in [Sha56] under the term zero error capacity.*

Definition 5.4. *We say that a channel is* ***synchronized*** *if the transmission of an input symbol of length* 1 *always results in an output of length* 1. *As a consequence this means that an observer given the input and the output of a block transmission can always tell which input symbol lead to which output symbol for the channel.*

Remark 5.2. *As mentioned in the introduction of the thesis we are covering the Z-channel (Chapter 6), specific unidirectional channels (Chapter 7) and insertion-deletion channels (Chapter 8). The first two classes belong to the family of synchronized channels, whereas the insertion-deletion channels do not.*

Synchronized channels with finite input and output alphabets are often specified by bipartite graphs.

Definition 5.5 (Synchronized channel/bipartite graph)**.** *A synchronized channel corresponds to a bipartite graph in the following way. Let* $\mathcal{X}$ *denote the set of possible input symbols and let* $\mathcal{Y}$ *denote the set of possible output symbols. Then if it is possible that the channel maps input symbol* $i \in \mathcal{X}$ *to output symbol* $j \in \mathcal{Y}$, *the pair* (i, j) *is part of the set of edges* $\mathcal{E} \subset \mathcal{X} \times \mathcal{Y}$. *Conversely, a bipartite graph between input symbols* $\mathcal{X}$ *and* $\mathcal{Y}$ *defines a synchronized channel.*

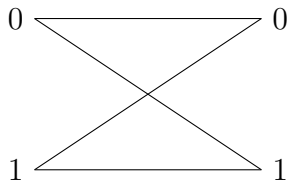

Figure 5.3.2: Symmetric channel Γ_S^2

Synchronized channels with finite input and output alphabets correspond to their respective bipartite graphs by a one to one mapping. Therefore, in this thesis we frequently speak about the graphs when we mean the respective channels and vice versa.

Definition 5.6 (Symmetric channel)**.** *A discrete channel is called* ***symmetric*** *if the channel can map every symbol of its input alphabet to every symbol of its output alphabet, i.e. for the set of edges $\mathcal{E}$ in its graph respresentation it holds that $\mathcal{E} = \mathcal{X} \times \mathcal{Y}$. We denote this channel by Γ_S^q.*

For the symmetric channel with binary input and output alphabets Γ_S^2, depicted in Fig. 5.3.2, the capacity error function has been established by Berlekamp [Ber64; Ber68] and Zigangirov [Zig76].

Theorem 5.1. *The capacity error function of Γ_S^2 with noiseless feedback is*

$$C_2^f(\Gamma_S^2, \tau) = \begin{cases} 1 - h(\tau), & \text{for } 0 \leq \tau \leq (3-\sqrt{5})/4, \\ (1-3\tau)\log\left(\frac{1+\sqrt{5}}{2}\right), & \text{for } (3-\sqrt{5})/4 < \tau \leq 1/3, \\ 0, & \text{otherwise .} \end{cases} \tag{5.5}$$

Definition 5.7 (Asymmetric channel)**.** *An* ***asymmetric channel*** *is a discrete channel whose specifying bipartite graph is not the full graph in the sense that the set of edges $\mathcal{E} \neq \mathcal{X} \times \mathcal{Y}$.*

6
The Z-Channel

Abstract

In this chapter, we compute lower and upper bounds on the capacity error function of the Z-channel with noiseless feedback. We consider a combinatorial error model where the maximal number of errors inflicted by an adversary is proportional to the number of channel uses within a transmission block in the asymptotic setting. We present an encoding strategy having an implementation complexity scaling linear with the blocklength. Without feedback, it is known that the capacity error function goes to zero when the error fraction $\tau \geq 1/4$. The coding scheme with feedback that we are presenting in this chapter is strictly positive for any fraction of errors $\tau < 1$. Additionally, we present an upper bound on the capacity error function for the Z-channel.

This chapter is based on [DLMP22] published in the IEEE Transactions on Information Theory. Parts of this work furthermore have been presented at the International Computing and Combinatorics Conference 2020 [DLMP20]. The author order within these papers has been chosen to be alphabetical. The coauthors of these works are all senior researchers and significant contributions to both mentioned works have been made by the author of this thesis.

6.1 Introduction

In optical communications and other digital transmission systems the ratio between the error probabilities of type $1 \to 0$ and $0 \to 1$ can be large [Bla94]. Practically, one can assume that only one type of error can occur. These channels are called asymmetric. In this chapter, we discuss the problem of finding coding strategies for the Z-channel with feedback. Throughout this chapter the Z-channel channel Γ_Z^2 that we are considering is specified in Fig. 6.1.1. The channel is of asymmetric nature because it permits an error $1 \to 0$, whereas it prohibits an error $0 \to 1$.

As already mentioned in Chapter 5, we are considering only the combinatorial setting where we limit the fraction of erroneous symbols by a fraction $\tau := \frac{t}{n}$. This is in contrast to the probabilistic setting, in which the error probability of the channel is fixed. Feedback codes achieving the Z-channel capacity (probabilistic setting) are

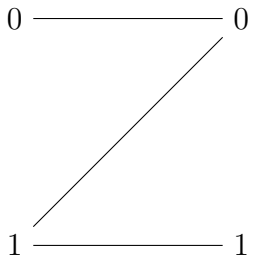

Figure 6.1.1: Z-channel Γ_Z^2

considered in [TABB08]. The figure of merit examined in this work also used in [ADL06] is the capacity error function, written as $C_2^f(\Gamma_Z^2, \tau)$ which is specified in Definition 5.3.

The problem of finding encoding strategies for the Z-channel using noiseless feedback is equivalent to a variation of Ulam's game, the half-lie game. The first appearance of the half-lie game occurs in [RMK+80]. In this game for two players one player, referred to as Paul, tries to find an element $x \in \mathcal{M}$ by asking n yes-no questions which are of the form: Is $x \in A$ for some $A \subseteq \mathcal{M}$? The other player, the responder Carole, knows x and is allowed to lie at most t times if the correct answer to the question is yes. In comparison to the original Ulam game [Ula91], Carole is not allowed to lie if the correct answer is no. Before Ulam proposed the game it was already described by Berlekamp [Ber68] and by Renyi [Ren61]. For a survey of results see [Cic13]. It is known that for fixed t asymptotically it holds that $M_2(\Gamma_Z^2, n, t) \sim 2^{n+t} t! n^{-t}$ as $n \to \infty$ for Paul to win the half-lie game. First this was shown for $t = 1$ in [CM00] and later generalized in [DS04; SY03] for arbitrary t. Due to the equivalence of the half-lie game and the coding problem with feedback for the Z-channel, the coding problem has been solved for an arbitrary but fixed number of errors for the asymptotic case when n goes to infinity.

Unlike studies in [SY03; DS04; CM00], we discuss the case when t is linear with n, i.e. $t = \tau n$ for some constant $0 \leq \tau \leq 1$. In this setting, it is natural to investigate the exponential growth of the quantity $M_2(\Gamma_Z^2, n, t)$. By random arguments, it was proved in [D'y75] that $C_2^f(\Gamma_Z^2, \tau) > 0$ for $\tau < 1/2$. In [DLM21] a feedback strategy based on the rubber method [ADL06] was introduced to find an encoding strategy achieving a positive asymptotic rate for any $\tau < 1/2$. The results in this chapter will show that significant improvements are possible. In particular we show that $C_2^f(\Gamma_Z^2, \tau)$ is indeed strictly positive for any $\tau < 1$.

6.1.1 Outline

The remainder of the chapter is organized as follows. In Section 6.2, we provide an upper bound on $C_2^f(\Gamma_Z^2, \tau)$. Then we provide an encoding algorithm achieving a positive asymptotic rate for any fraction of errors $\tau < 1$ which is the main result

presented in this chapter (Theorem 6.2). In Section 6.4 we set our results in context to known results in the literature and constitute our conjecture that the established lower bound achieved by the presented coding scheme is tight. Finally, we conclude the results of this chapter in Section 6.5.

6.2 Upper Bound on $C_2^f(\Gamma_Z^2, \tau)$

In this section we establish an upper bound on $C_2^f(\Gamma_Z^2, \tau)$. This upper bound is close to our lower bound for small values of τ. We make use of an approach similar to the one in [SY03]. We take a successful encoding strategy and consider only messages $m \in \mathcal{M}$ such that any output sequence in $\mathcal{Y}_t^n(m)$ has a relatively large Hamming weight. For those messages, it is possible to derive a good lower bound on the size of $\mathcal{Y}_t^n(m)$. The upper bound on the size of the set of possible messages is then obtained by a sphere-packing argument.

Theorem 6.1. *For any τ, $0 < \tau < 1$, we have $R(\tau) \leq \overline{R}(\tau)$, where $\overline{R}(\tau)$ is defined as*

$$\overline{R}(\tau) =: \min_{\substack{0 \leq \tau' \leq \tau \\ 0 \leq \omega \leq 1-\tau'}} \max\Big\{ f_1(\tau, \tau', w), f_2(\tau, \tau', w) \Big\},$$

and

$$\begin{aligned} f_1(\tau, \tau', w) &:= (1-\tau') h\left(\max\left\{ \frac{\omega}{1-\tau'}, \frac{1}{2} \right\} \right) \\ &\quad -(\omega + \tau - \tau') h\left(\frac{\tau - \tau'}{\omega + \tau - \tau'} \right), \\ f_2(\tau, \tau', w) &:= (1-\tau') h\left(\min\left\{ \frac{\omega}{1-\tau'}, \frac{1}{2} \right\} \right). \end{aligned}$$

Proof. Let $\mathcal{M}$ denote the set of possible messages. Let

$$c(m, y^{n-1}) = (c_1(m), c_2(m, y^1), \ldots, c_n(m, y^{n-1}))$$

be a fixed successful encoding algorithm with

$$c_i(m, y^{i-1}) : \mathcal{M} \times \{0,1\}^{i-1} \to \{0,1\}$$

for all $1 \leq i \leq n$ and t be the maximum number of asymmetric errors. Recall that for a given message $m \in \mathcal{M}$, the set of output sequences, written as $\mathcal{Y}_t^n(m)$, is defined by equation (5.3) in this case specifically with $d = d_H$ and $\mathcal{Y} = \{0,1\}$. Let l and w be fixed integers such that $0 \leq l \leq t$ and $0 \leq w \leq n - l$. A message $m \in \mathcal{M}$ is said to be (l, w)-*good* if the Hamming weight of any $y^n \in \mathcal{Y}_{t,l}^n(m)$ is at least $w+1$, where $\mathcal{Y}_{t,l}^n(m)$ is the set of output sequences for the message m that have zeros at their last l positions,

i.e., $\mathcal{Y}_{t,l}^n(m) := \{y^n \in \mathcal{Y}_t^n(m) : \ y_i = 0 \text{ for } i \in \{n-l+1, \ldots, n\}\}$. Since $l \leq t$, for any message $m \in \mathcal{M}$, the channel can inflict errors in such a way that it forces the output sequence to contain zeros in its last l positions. Hence, $|\mathcal{Y}_{t,l}^n(m)| \geq 1$ for any $m \in \mathcal{M}$. Let $\mathcal{G} \subset \mathcal{M}$, denote the set of (l, w)-good messages and m be a fixed (l, w)-good message. In what follows, we shall prove that the size of $\mathcal{Y}_{t,l}^n(m)$ is bounded below as

$$|\mathcal{Y}_{t,l}^n(m)| \geq \binom{w+t-l}{t-l}. \tag{6.1}$$

Let I denote an arbitrary subset of $t-l$ elements of the set $[w+t-l]$. Now we construct $y^n = y^n(m, I) \in \{0,1\}^n$ using an algorithm and show that the constructed y^n is a potential output sequence, i.e., $y^n \in \mathcal{Y}_{t,l}^n(m)$. In the following algorithm, $u^n = (u_1, \ldots, u_n) = u^n(m, I)$ denotes the transmitted sequence, j is the number of already transmitted symbols, w_j is the Hamming weight of the sequence $u^j = (u_1, \ldots, u_j)$ and y^0 denotes the empty string.

Input: Message m, encoding strategy $c(m, y^{n-1})$ and set $I \subset [w+t-l]$, $|I| = t-l$.
Initialisation: Let $j \leftarrow 0$, $w_j \leftarrow 0$.
Step 1: Set $u_{j+1} \leftarrow c_{j+1}(m, y^j)$. Update $j \leftarrow j+1$ and $w_j \leftarrow w_{j-1} + u_j$. If $u_j = 1$ and $w_j \in I$, then $y_j \leftarrow 0$. Otherwise $y_j \leftarrow u_j$. If $j < n-l$, repeat Step 1. Otherwise, go to Step 2.
Step 2: Set $u_{j+1} \leftarrow c_{j+1}(m, y^j)$. Update $j \leftarrow j+1$, $w_j \leftarrow w_{j-1} + u_j$ and $y_j \leftarrow 0$. If $j < n$, repeat Step 2. Otherwise, exit the algorithm.
Output: Sequence $y^n = (y_1, \ldots, y_n)$.

The sequence y^n is a potential output of the Z-channel if the encoder transmits the message m since $y_j \leq u_j = c_j(m, y^{j-1})$ for all $j \in [n]$. The number of errors at Step 1 is $d_H(y^{n-l}, u^{n-l}) \leq |I| = t-l$. The output symbols at Step 2 are zeros and the number of errors at Step 2 is at most the remaining blocklength l. Hence, the total number of errors $d_H(y^n, u^n) \leq t-l+l = t$, which implies that $y^n \in \mathcal{Y}_{t,l}^n(m)$.

We claim that $w_H(u^{n-l}) = w_{n-l} \geq w+t-l+1$ and, thus, the number of errors at Step 1 is $d_H(u^{n-l}, y^{n-l}) = t-l$. Note that $d_H(u^{n-l}, y^{n-l})$ is the number of elements in I that are less than or equal to w_{n-l}. By definition of the set I, the s^{th} smallest element in I is at most $w+s$. Thus $d_H(u^{n-l}, y^{n-l})$ is at least the maximum $s \in [t-l]$ such that $w+s \leq w_{n-l}$, i.e., $d_H(u^{n-l}, y^{n-l}) \geq \min(t-l, w_{n-l}-w)$. Since the message m is (l, w)-good, it holds that $w+1 \leq w_H(y^n) = w_H(y^{n-l}) = w_H(u^{n-l}) - d_H(y^{n-l}, u^{n-l}) \leq w_{n-l} - \min\{t-l, w_{n-l}-w\}$. By comparing the leftmost side and the rightmost side of the above inequality, we derive that the minimum in $\min\{t-l, w_{n-l}-w\}$ is attained at the first argument since, otherwise, it holds that $w+1 \leq w$. Therefore, $t-l < w_{n-l}-w$ or $w_{n-l} \geq w+t-l+1$.

We now check that $y^n(m, I) \neq y^n(m, K)$ for any distinct $K, I \subset [w+t-l]$, $|K| = |I| = t-l$. Let f_I (f_K) be the smallest element in I (K) that does not appear in K (I). Without loss of generality, consider the case $f_I < f_K$. Since exactly $t-l$ errors occurred

at Step 1 when receiving $y(m, I)$, there is a moment $j_0 \in [n-l]$ such that $u_{j_0} = 1$ and $w_{j_0} = f_I \in I$. Then $y_{j_0}(m, I) = 0 \neq 1 = y_{j_0}(m, K)$ and $y_i(m, I) = y_i(m, K)$ for all $i \in [j_0 - 1]$. This implies the required bound (6.1) since the number of different $(t-l)$-element subsets of the set $[w+t-l]$ is exactly $\binom{w+t-l}{t-l}$.

Recall that the fixed strategy is successful and, thus, the sets $\mathcal{Y}_{t,l}^n(m)$ are pairwise disjoint for all $m \in \mathcal{M}$. The number of binary sequences of length $n-l$ with at least $w+1$ ones is $\sum_{w'=w+1}^{n-l} \binom{n-l}{w'}$ and, thus, $\sum_{m \in \mathcal{G}} |\mathcal{Y}_{t,l}^n(m)| \leq \sum_{w'=w+1}^{n-l} \binom{n-l}{w'}$. By (6.1), we have that

$$|\mathcal{G}| \leq \frac{\sum_{w'=w+1}^{n-l} \binom{n-l}{w'}}{\binom{w+t-l}{t-l}}.$$

Recall that $|\mathcal{Y}_{t,l}^n(m)| \geq 1$ for all messages $m \in \mathcal{M}$. From the definition of messages that are not (l, w)-good, we obtain $\sum_{m \in \mathcal{M} \setminus \mathcal{G}} |\mathcal{Y}_{t,l}^n(m)| \leq \sum_{w'=0}^{w} \binom{n-l}{w'}$. Hence,

$$|\mathcal{M} \setminus \mathcal{G}| \leq \sum_{w'=0}^{w} \binom{n-l}{w'}.$$

Finally, by optimizing over parameters l and w, we can upper bound the value $M(n, t)$ by

$$\min_{0 \leq l \leq t} \min_{1 \leq w \leq n-l} \left\{ \frac{\sum_{w'=w+1}^{n-l} \binom{n-l}{w'}}{\binom{w+t-l}{t-l}} + \sum_{w'=0}^{w} \binom{n-l}{w'} \right\}.$$

By letting $n \to \infty$ and setting $\tau' := l/n$ and $\omega := w/n$, we get that $R(\tau)$ is bounded above by

$$\min_{0 \leq \tau' \leq \tau} \min_{0 \leq \omega \leq 1-\tau'} \max \{ f_1(\tau, \tau', \omega),\ f_2(\tau, \tau', \omega) \}$$

as claimed in this theorem. ■

6.3 Lower Bound on $C_2^f(\Gamma_Z^2, \tau)$

In this section we give a successful encoding strategy for the Z-channel. This gives a lower bound on the capacity error function for the Z-channel capable of correcting a fraction of τ asymmetric errors $C_2^f(\Gamma_Z^2, \tau)$.

Theorem 6.2. *For any τ, $0 \leq \tau \leq 1$, we have that $C_2^f(\Gamma_Z, \tau) \geq \underline{R}(\tau)$, where $\underline{R}(\tau)$ is defined as*

$$\begin{aligned} \underline{R}(\tau) &:= (1+\tau)\left(1 - h\left(\frac{\tau}{1+\tau}\right)\right) \\ &= (1+\tau) - (1+\tau)\log_2(1+\tau) + \tau \log_2 \tau \ . \end{aligned}$$

6.3.1 Encoding Strategy

At the start of the message transmission the receiver has only knowledge about the set of messages $\mathcal{M}$ and the encoding strategy the sender deploys. The sender's task is to communicate a message $m \in \mathcal{M}$ to the receiver. The sender does so by applying an encoding strategy which reduces the number of *eligible* messages from the receiver's viewpoint until only the message m is left. The encoding algorithm we provide divides the number of channel uses n into subblocks. Therefore, the encoding procedure is potentially divided into several steps. We denote the set of eligible messages from the receiver's perspective after the i^{th} step as $\mathcal{M}_{i+1}$ with $M_{i+1} := |\mathcal{M}_{i+1}|$, the number of remaining channel uses as n_{i+1} and the number of remaining errors available to the adversary by t_{i+1}.

In the following the algorithm depicted in Figure 6.3.1 is described. Before every step, the sender (as well as the receiver) checks the following two properties

$$t_i = 0 \tag{6.2}$$

and

$$|\mathcal{M}_i| \leq n_i - t_i + 1 \ . \tag{6.3}$$

For the sender checking the aforementioned conditions before each step is possible due to the feedback. Depending on which of the conditions (6.2)-(6.3) hold, the sender chooses one out of three algorithms for encoding. The receiver is able to keep track of the amount of already occurred errors due to the specifics of the Partitioning Algorithm which is specified in detail later in this section. This information is essential because it tells the decoder when the encoder switches to a different algorithm.

If both conditions (6.2)-(6.3) do not hold, then the sender makes use of Partitioning Algorithm. This strategy tries to limit the set of eligible messages by dividing the message space into subsets and sending the index of the subset containing the message. After this subblock transmission the sender and the receiver examine the conditions (6.2)-(6.3) and check whether the encoding and decoding strategies have to be adjusted for the remainder of the block.

If property (6.2) is violated and property (6.3) holds, then the sender uses the Weight Algorithm for information transmission in the remaining channel uses.

If the condition (6.2) is true, then the sender applies the Uncoded Algorithm for information transmission in the remaining channel uses. Below we describe the three algorithms required for our encoding strategy.

Partitioning Algorithm: This algorithm relies on the specific choice of positive fixed integers δ and p with $\delta > p$ (see Lemma 6.1). We partition the message space

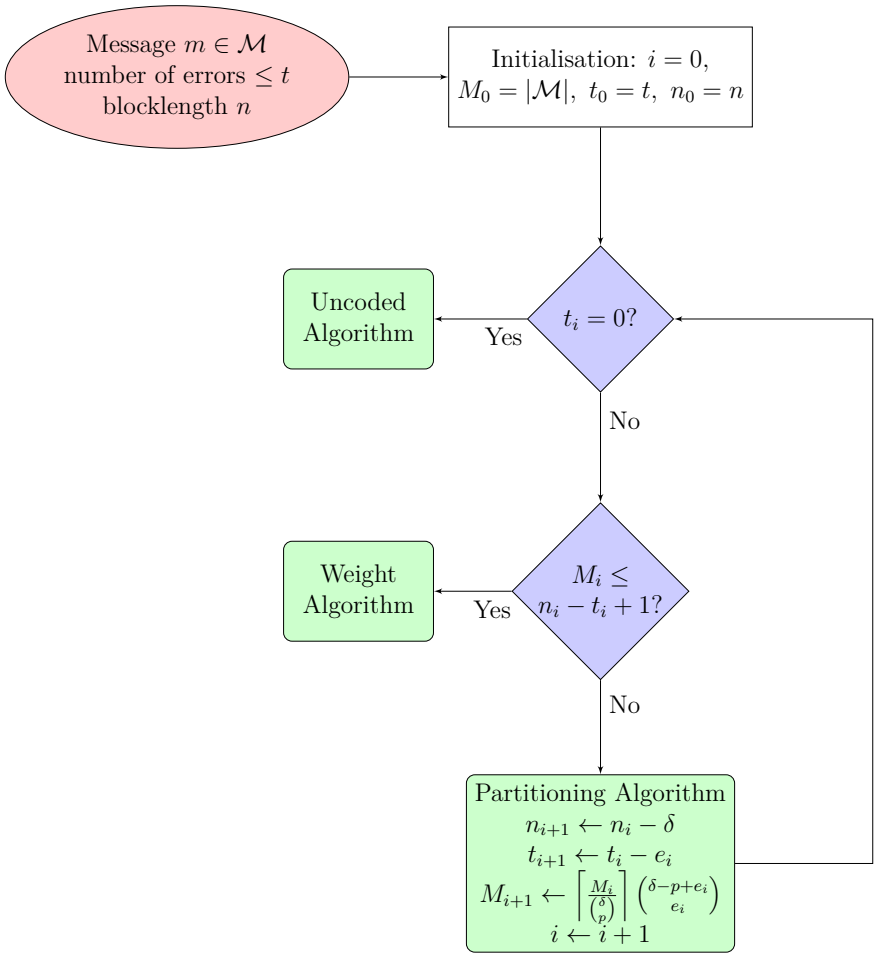

Figure 6.3.1: Encoding algorithm for transmission over the Z-channel

before the i^{th} step $\mathcal{M}_i$ into $\binom{\delta}{p}$ subsets $\mathcal{M}_{i,j}$ of almost equal sizes

$$\mathcal{M}_i = \bigcup_{j=1}^{\binom{\delta}{p}} \mathcal{M}_{i,j} \ .$$

The size of each subset is either $\left\lceil M_i / \binom{\delta}{p} \right\rceil$, or $\left\lfloor M_i / \binom{\delta}{p} \right\rfloor$. The exact way in which the message space is to be partitioned is to be agreed between the sender and the receiver before the data transmission. Then the sender finds the index of the group containing the message and transmits this index using a subblock of length δ containing p ones. In this way the receiver can determine the number of errors inflicted by the channel within this subblock by counting the number of ones. There are $p+1$ possible cases depending on the number of errors e_i within the respective subblock of the i^{th} step.

If $e_i = p$ errors occur, the received subblock consists of δ zeros and the set of possible messages remains unchanged from the receiver's point of view, i.e. $\mathcal{M}_{i+1} = \mathcal{M}_i$, $n_{i+1} = n_i - \delta$ and $t_{i+1} = t_i - p$.

When $e_i < p$ errors occur, there are $\binom{\delta-p+e_i}{e_i}$ subsets of messages $\mathcal{M}_{i,k}$ that are consistent with the outcome of the Z-channel. $\mathcal{M}_{i+1}$ is then equal to the union of these subsets. Therefore, the set of eligible messages in accordance with the received δ symbols is reduced and we have $M_{i+1} \leq \left\lceil M_i / \binom{\delta}{p} \right\rceil \binom{\delta-p+e_i}{e_i}$. Moreover, the receiver and the sender obtain $n_{i+1} = n_i - \delta$ and $t_{i+1} = t_i - e_i$.

Weight Algorithm: We order the messages within the set of eligible messages, denoted by $\mathcal{M}'$, by enumerating them: $\mathcal{M}' = \{m_0, m_1, \ldots, m_{|\mathcal{M}'|-1}\}$. The sender would like to transmit one of the messages in $\mathcal{M}'$, say m_r, to the receiver by using the channel n' times. To this end, the sender transmits a 1 over the channel until a 1 is received exactly r times. This happens at some point if a sufficient amount of channel uses is considered because the number of errors is limited. We denote this limit as t'. After that, the sender transmits 0-symbols which cannot be disturbed by the Z-channel. The receiver finds the Hamming weight w of the received sequence of length n' and outputs the message m_w. This strategy is successful, i.e., $m_r = m_w$, if $|\mathcal{M}'| \leq n' - t' + 1$.

Uncoded Algorithm: We denote the ordered set of eligible messages as $\mathcal{M}' = \{m_0, m_1, \ldots, m_{|\mathcal{M}'|-1}\}$. The sender's task is to send one of the messages, say m_r, to the receiver by using the channel n' times. In order to do so, it sends the (standard) binary representation of the index r over the channel. This strategy is successful if the sender is allowed to use the channel at least $\lceil \log_2 |\mathcal{M}'| \rceil$ times and no errors occur.

Next we prove a lower bound on the maximal cardinality of the set of possible messages $\mathcal{M}$.

6.3.2 Analysis of the Proposed Algorithm

Lemma 6.1. *Let k be the maximal number of partitioning steps, let δ be the length of a partitioning subblock having Hamming weight p. Let t denote the maximal number of asymmetric errors and let ε be an arbitrary real number such that $0 < \varepsilon < 1 - \frac{p}{\delta}$. Then we obtain that*

$$\gamma_e := (1-\varepsilon)\frac{\binom{\delta}{p}}{\binom{\delta-p+e}{e}} > 1, \quad \forall e \in \{0, \ldots, p-1\}.$$

We define $\gamma_p := 1$, $A := \left\lceil \binom{\delta}{p}/\varepsilon \right\rceil$ and the set $\mathcal{S}_{(k,t,p)}$ is defined as

$$\left\{(k_0, \ldots, k_p) \in \mathbb{N}_0^{p+1} : \sum_{i=0}^{p} k_i = k, \sum_{i=0}^{p} i k_i \leq t + p\right\}.$$

Then for any choice of t and k such that $\delta k \geq t$, it holds

$$M_2(A + \delta k, t) \geq \left\lfloor A \min_{\mathcal{S}_{(k,t,p)}} \prod_{e=0}^{p} \gamma_e^{k_e} \right\rfloor. \tag{6.4}$$

In particular, it follows that

$$M_2(A + \delta k, t) \geq A \frac{(1-\varepsilon)^k \binom{\delta}{p}^k}{\binom{\delta k - pk + t + p}{t+p}} - 1. \tag{6.5}$$

Remark 6.1. *The set $\mathcal{S}_{(k,t,p)}$ can be interpreted as the set of all possible error distributions during the partitioning steps that may happen when the sender transmits the message. Concretely, a tuple $(k_0, k_1, \ldots, k_p)$ specifies that during the encoding procedure of the message there have been k_i partitioning subblocks for which exactly i errors have occurred. We also note that the set $\mathcal{S}_{(k,t,p)}$ is of finite cardinality and includes the tuple $(k, 0, \ldots, 0)$. Thus, the minimization in* (6.4) *is well defined.*

Proof. We prove the statement by showing that the encoding strategy described in Section 6.3.1 is successful for a message set of size equal to the right-hand side of (6.4). We first consider the case that $t = 0$. In this case the Uncoded Algorithm is used by the encoder. Observe that the minimum on the right and side of (6.4) is upper bounded by $A\gamma_0^k \leq A\binom{\delta}{p}^k \leq A2^{\delta k}$. Since the blocklength is $A + \delta k$ a message size of $2^{A+\delta k}$ is possible which is larger than $A2^{\delta k}$.

It remains to show inequality (6.4) for the case that $t \neq 0$. We do this by using induction over k. Since $\delta k \geq t$ the base case of the induction is $k = \lceil t/\delta \rceil$. In this case the minimum on the right-hand side of (6.4) is equal to $\lfloor A \rfloor$ and attained for

$(k_0, k_1, \ldots, k_p) = (0, \ldots, 0, k)$ since $\gamma_0 > \gamma_1 > \ldots > \gamma_p = 1$ and

$$\sum_{i=0}^{p} k_i = k, \quad \sum_{i=0}^{p} i k_i = p\lceil t/\delta \rceil \leq pt/\delta + p \leq t + p.$$

Due to the specification of the encoder the Weight Algorithm is to be used. Indeed, condition (6.3) holds because

$$n - t + 1 = A + \delta k - t + 1 \geq A + 1 \geq A = \lfloor A \rfloor$$

and the encoder is able to handle the required cardinality of the message set for the base case.

To complete the proof of the lemma we have to show the inductive step. Since we are not in a base case anymore, we have $k > \lceil t/\delta \rceil$. We have to prove that the cardinality of the message set is at least

$$M := \left\lfloor A \min_{\mathcal{S}_{(k,t,p)}} \prod_{i=0}^{p} \gamma_i^{k_i} \right\rfloor$$

using the inductive hypothesis. According to the above algorithm, the encoder checks whether the two conditions (6.2)-(6.3) hold. The case that (6.2) holds corresponds to the case that $t = 0$ which has already been discussed. If condition (6.3) holds, the encoder uses the Weight Algorithm to successfully transmit the message.

If both conditions are failed, the sender uses the Partitioning Algorithm. In that case it remains to show that for any $e \in \{0, 1, \ldots, p\}$ (the number of errors in the first block of δ bits), the sender is able to communicate the message m by using the remaining $A + \delta(k-1)$ channel uses while at most $t - e$ errors are allowed for the remainder of the block. Notice that there are at most

$$\begin{cases} \left\lceil \frac{M}{\binom{\delta}{p}} \right\rceil \binom{\delta - p + e}{e}, & \text{for } e \in \{0, 1, \ldots, p-1\} \\ M, & \text{for } e = p \end{cases}$$

eligible messages left after the partitioning step. By the inductive hypothesis we are able to transmit

$$\left\lfloor A \min_{\mathcal{S}_{(k-1,t-e,p)}} \prod_{i=0}^{p} \gamma_i^{k_i} \right\rfloor$$

messages for this setup if the condition $\delta(k-1) \geq t - e$ holds. The latter inequality follows because $k > \lceil t/\delta \rceil$, which implies $k \geq 1 + \lceil t/\delta \rceil$ or $\delta(k-1) \geq \delta\lceil t/\delta \rceil \geq t$. To

complete the inductive step we show that

$$\left\lceil \frac{M}{\binom{\delta}{p}} \right\rceil \binom{\delta - p + e}{e} \leq \left\lfloor A \min_{\mathcal{S}_{(k-1,t-e,p)}} \prod_{i=0}^{p} \gamma_i^{k_i} \right\rfloor, \tag{6.6}$$

for all $e \in \{0, 1, \ldots, p-1\}$. Examining the left hand side of the inequality we observe

$$\begin{aligned} \left\lceil \frac{M}{\binom{\delta}{p}} \right\rceil \binom{\delta - p + e}{e} &\leq \frac{Mp!(\delta - p + e)!}{\delta! e!} + \binom{\delta - p + e}{e} \\ &\overset{(a)}{\leq} \frac{Mp!(\delta - p + e)!}{\delta! e!} + A\varepsilon \\ &\overset{(b)}{=} \frac{M(1-\varepsilon)}{\gamma_e} + A\varepsilon, \end{aligned} \tag{6.7}$$

where we used the property $A\varepsilon = \lceil \binom{\delta}{p} / \varepsilon \rceil \varepsilon \geq \binom{\delta - p + e}{e}$ for any $e \in \{0, 1, \ldots, p\}$ in (a) and $\gamma_e = (1-\varepsilon)\frac{\delta! e!}{p!(\delta - p + e)!}$ in (b). We note that

$$M = \lfloor A \min_{\mathcal{S}_{(k,t,p)}} \prod_{i=0}^{p} \gamma_i^{k_i} \rfloor \leq \lfloor A\gamma_e \min_{\mathcal{S}_{(k-1,t-e,p)}} \prod_{i=0}^{p} \gamma_i^{k_i} \rfloor. \tag{6.8}$$

The latter inequality holds because

$$\gamma_e \min_{\mathcal{S}_{(k-1,t-e,p)}} \prod_{i=0}^{p} \gamma_i^{k_i} \geq \min_{\mathcal{S}_{(k,t,p)}} \prod_{i=0}^{p} \gamma_i^{k_i}$$

as we get the left-hand side of the inequality from the right-hand side by adding the additional constraint that $k_e \geq 1$ to the minimization. Thus, combining the inequalities (6.7)-(6.8), we get

$$\begin{aligned} \left\lceil \frac{M}{\binom{\delta}{p}} \right\rceil \binom{\delta - p + e}{e} &\leq A(1-\varepsilon) \min_{\mathcal{S}_{(k-1,t-e,p)}} \prod_{i=0}^{p} \gamma_i^{k_i} + A\varepsilon \\ &\overset{(c)}{\leq} A \min_{\mathcal{S}_{(k-1,t-e,p)}} \prod_{i=0}^{p} \gamma_i^{k_i}. \end{aligned}$$

To prove (c), we observe that $\gamma_i \geq 1$ for all i. As we compare the integer on the left-hand side with a real number on right-hand side of the inequality, we can apply the floor operation to the latter. This proves (6.6) and completes the proof of the inductive step.

It remains to show (6.5). Suppose that the minimum in (6.4) is attained with

$k_0 = k'_0, k_1 = k'_1, \ldots, k_p = k'_p$ such that

$$\sum_{i=0}^{p} k'_i = k, \quad \sum_{i=0}^{p} i k'_i =: t' \le t + p.$$

Then we derive

$$M = \left\lfloor A \prod_{i=0}^{p} \gamma_i^{k'_i} \right\rfloor \ge A(1-\varepsilon)^k \prod_{i=0}^{p} \frac{\binom{\delta}{p}^{k'_i}}{\binom{\delta-p+i}{i}^{k'_i}} - 1 = A \frac{(1-\varepsilon)^k \binom{\delta}{p}^k}{\prod_{i=0}^{p} \binom{\delta-p+i}{i}^{k'_i}} - 1$$
$$\overset{(d)}{\ge} A \frac{(1-\varepsilon)^k \binom{\delta}{p}^k}{\binom{\delta k - pk + t'}{t'}} - 1 \overset{(e)}{\ge} A \frac{(1-\varepsilon)^k \binom{\delta}{p}^k}{\binom{\delta k - pk + t + p}{t+p}} - 1,$$

where the property $\binom{u}{v}\binom{w}{z} \le \binom{u+w}{v+z}$ yields (d) and the monotonicity of the function $\binom{u+x}{x}$ in x implies (e). ■

Now we use Lemma 6.1 to prove Theorem 6.2.

Proof of Theorem 6.2. Let us fix some positive $\tau < 1$. For any $\varepsilon_R > 0$ and small enough $\varepsilon_\tau > 0$, we shall prove the existence of a code of an arbitrarily large blocklength and code rate at least $\underline{R}(\tau) - \varepsilon_R$ capable of correcting a fraction $\tau - \varepsilon_\tau$ of errors.

In what follows, we vary positive integers k and δ with $k > \delta$. Define $t = \lceil \tau k \delta \rceil$, $p = \lfloor \delta(1/2 + \tau/2) \rfloor$, $A = \lceil \binom{\delta}{p} / \varepsilon \rceil$ and $n = A + \delta k$, where the real parameter ε is fixed and satisfies

$$0 < \varepsilon < \frac{1-\tau}{2} \le 1 - p/\delta,$$
$$0 < \underline{R}(\tau) + \log_2(1-\varepsilon) - 3\varepsilon_\tau. \tag{6.9}$$

Let δ_0 be such that for any $\delta \ge \delta_0(\varepsilon_\tau, \tau)$ and $k \ge \delta$, we have

$$\binom{\delta}{p} \ge 2^{\delta\left(h\left(\frac{1+\tau}{2}\right) - \varepsilon_\tau\right)} \tag{6.10}$$

and

$$\binom{\delta k - pk + t + p}{t+p} \le 2^{\delta k \frac{1+\tau}{2}\left(h\left(\frac{2\tau}{1+\tau}\right) + \varepsilon_\tau\right)}. \tag{6.11}$$

To prove the existence of such δ_0, we note that

$$\lim_{\delta\to\infty} \frac{p}{\delta} = \lim_{\delta\to\infty} \frac{\lfloor \delta(1/2 + \tau/2) \rfloor}{\delta} = \frac{1+\tau}{2}, \tag{6.12}$$

and for $k \geq \delta$,

$$\lim_{\delta \to \infty} \frac{t+p}{\delta k - pk + t + p} = \frac{2\tau}{1+\tau},$$

and for any integers $u > v \geq 1$, the binomial coefficient $\binom{u}{v}$ satisfies

$$\sqrt{\frac{u}{8v(u-v)}} 2^{uh(v/u)} \leq \binom{u}{v} \leq \sqrt{\frac{u}{2\pi v(u-v)}} 2^{uh(v/u)}. \tag{6.13}$$

Then we take $k_0 = k_0(\delta, \tau, \varepsilon_\tau)$ such that for any $k \geq k_0$, the fraction of errors

$$\frac{t}{n} = \frac{t}{A + \delta k} \geq \tau - \varepsilon_\tau$$

and the blocklength

$$n = A + \delta k \leq \delta k(1 + \varepsilon_\tau) \tag{6.14}$$

and

$$\frac{(1-\varepsilon)^k \binom{\delta}{p}^k}{\binom{\delta k - pk + t + p}{t+p}} \geq 2.$$

The latter can be achieved because of the choice of ε in (6.9) and large enough δ in (6.10)-(6.11). By Lemma 6.1, there exists a feedback error-correcting code with blocklength $n = A + \delta k$ for a message space of size

$$M \geq A \frac{(1-\varepsilon)^k \binom{\delta}{p}^k}{\binom{\delta k - pk + t + p}{t+p}} - 1 \geq \frac{(1-\varepsilon)^k \binom{\delta}{p}^k}{2\binom{\delta k - pk + t + p}{t+p}}, \tag{6.15}$$

capable of correcting t errors when transmitted through the Z-channel. Thus, combining (6.10)-(6.15) yields

$$\begin{aligned} R(\tau - \varepsilon_\tau) \geq \frac{\log_2 M}{A + \delta k} &\geq \frac{k \log_2(1-\varepsilon)}{\delta k(1+\varepsilon_\tau)} + \frac{\left(h\left(\frac{1+\tau}{2}\right) - \varepsilon_\tau - \frac{1+\tau}{2} h\left(\frac{2\tau}{1+\tau}\right) - \varepsilon_\tau\right)}{(1+\varepsilon_\tau)} - \frac{1}{\delta k(1+\varepsilon_\tau)} \\ &= (1+\tau)\log_2\left(\frac{2}{1+\tau}\right) + \tau \log_2 \tau - \varepsilon_R, \end{aligned}$$

where

$$\varepsilon_R \leq -\log_2(1-\varepsilon) + 3\varepsilon_\tau + \frac{1}{\delta k}.$$

As ε and ε_τ can be taken as small as needed and δ and k can be arbitrarily large, the statement of Theorem 6.2 follows. ■

6.3.3 Complexity of Encoding and Decoding Strategy

Practically, aside from the capacity error function it is also important that encoders and decoders are of small complexity. In this section we show that it is possible to choose the parameters within the coding scheme presented in Section 6.3.1 such that the complexity of the encoding algorithm is in the order $\mathcal{O}(n)$.

Theorem 6.3. *Let the maximal fraction of errors τ fulfill $0 \leq \tau \leq 1$ and let the block-length be n. Then there exist successful encoding and decoding schemes of complexity $\mathcal{O}(n)$ for the Z-channel with feedback that achieve the rate $\underline{R}(\tau)$ defined in Theorem 6.2 arbitrarily close as $n \to \infty$.*

Proof. Let M_i denote the size of the set $\mathcal{M}_i$. Recall that if $t_i > 0$ and $M_i > n_i - t_i + 1$ the encoder uses the Partitioning Algorithm to reduce the set of eligible messages. To prove Theorem 6.3 we identify the message m by an index within the set $\{1, \ldots, M_i\}$. Within the Partitioning Algorithm it was only specified that the set of eligible messages is to be split into subsets $\mathcal{M}_{i,j}$, of cardinality $\left\lceil M_i / \binom{\delta}{p} \right\rceil$ or $\left\lfloor M_i / \binom{\delta}{p} \right\rfloor$. We consider the set of messages $\mathcal{M}_i$ to be points on a line which are in ascending order according to the message index. We graphically illustrate the set of eligible messages as a set of discrete points on a straight line which we partition into $\binom{\delta}{p}$ segments $\mathcal{M}_{i,j}$ in the natural way depicted on Figure 6.3.2.

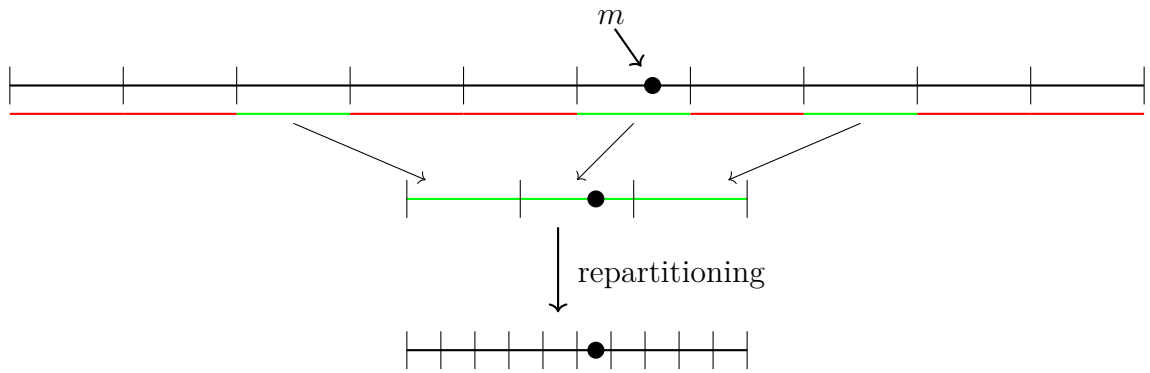

Figure 6.3.2: Partitioning algorithm

Since M_i may not be divisible by $\binom{\delta}{p}$ we have

$$r_i := M_i - \left\lfloor \frac{M_i}{\binom{\delta}{p}} \right\rfloor \binom{\delta}{p} \tag{6.16}$$

subsets of size $\left\lceil \frac{M_i}{\binom{\delta}{p}} \right\rceil$, in the following denoted as large segments, whereas the remaining ones are of size $\left\lfloor \frac{M_i}{\binom{\delta}{p}} \right\rfloor$. We define our partition in such a way that the larger segments

are located next to each other starting on the left hand side of the line. Furthermore, due to the fixed weight addressing of subsets within the partitioning algorithm each segment can be identified by a sequence of length δ containing exactly p ones. The addressing of the segments is in principle arbitrary but has to be known to sender and receiver. We make the simple choice to interpret the addresses as numbers in binary representation and order the addresses in ascending order from left to right.

To perform the partitioning, the encoder transmits the address of the segment containing the message m over the channel. If no error occurred the block is of Hamming weight p. For every error this Hamming weight is decreased by one. If we denote the number of errors by e there are $\binom{\delta-p+e}{e}$ segments in accordance with the received partitioning subblock. Those segments are addressed by the sequences which can be created by changing e zeros of the received partitioning subblock to ones. They are referred to in the following as *eligible* segments. The partitioning process is shown in Figure 6.3.2 where the eligible segments according to the channel output are underlined in green whereas the segments to be discarded are underlined in red. Due to the feedback the same information can be obtained by the sender, reducing the set of eligible messages. For the next partitioning step the eligible sets form the new message space. Notice that the order of the messages with respect to each other remains the same even if more than one partitioning step is performed. After each partitioning step the encoder checks whether properties (6.2) and (6.3) are still both not fulfilled. If this is the case the set of messages is repartitioned and another partitioning step is performed. Otherwise the encoder switches either to the Weight Algorithm or the Uncoded Algorithm, depending on the state of the conditions (6.2) and (6.3) (see also Figure 6.3.1).

During potentially numerous partitioning steps it is necessary to keep track of the cardinality of the set of eligible messages after the i-th partitioning step $\mathcal{M}_{i+1}$ as well as the relative position of m in within $\mathcal{M}_{i+1}$.

A possible way to do this is the creation of a lookup table with $\binom{\delta}{p}$ entries. The entries within this table contain the $\binom{\delta}{p}$ binary sequences in ascending order when being interpreted as numbers in binary representation. Utilizing the feedback this information is sufficient to compute the cardinality of the set of eligible messages $\mathcal{M}_{i+1}$ after the partitioning step as well as the index of m in $\mathcal{M}_{i+1}$. Indeed, by using the lookup table and investigating the received sequence (which is known to the encoder due to the feedback), the sender is able to obtain knowledge about all eligible segments after the i-th partitioning step. Furthermore, the sender knows the lengths of all eligible segments due to the predefined ordering of the segments and his ability to compute r_i using equation (6.16). This is sufficient to compute the cardinality of $\mathcal{M}_{i+1}$. Moreover, the encoder is able to obtain the relative index of m within the segment it is contained by using his knowledge of the lengths of the segments $\mathcal{M}_{i,j}$ and the relative index of m in $\mathcal{M}_i$. Eventually, for the computation of the index of m within $\mathcal{M}_{i+1}$ the encoder uses his knowledge about the ordering of the eligible

segments. To obtain the index of m within $\mathcal{M}_{i+1}$ it adds the length of the eligible segments left to the one containing m and the relative index of m within the segment it is contained in.

Next we describe how the decoder obtains knowledge about the message m from the received binary string. The decoder is able to compute the same lookup table for each partitioning step as the encoder does. The strategy of putting the large segments on the left hand side of the line is predefined and known by the decoder. Therefore, the received sequence is sufficient to compute the cardinality of $\mathcal{M}_{i+1}$ after each partitioning step. The number of errors within each received subblock is determined by its Hamming weight. The computation of $M_{i+1} = |\mathcal{M}_{i+1}|$ as well as the computation of the relative index of m within its containing segment require complexity $\mathcal{O}_\delta(1)$ for each partitioning step at the sender and the receiver side. The knowledge of M_{i+1} and t_{i+1} enables the decoder to find the point at which the encoder switches his strategy to either the Hamming weight algorithm or the Uncoded algorithm. The index of m within the eligible messages after the final partitioning step can be obtained in a straightforward manner according to the decoding of Weight or Uncoded algorithm. The partitioning steps can then be reversed in accordance to the output sequence to find the message m by plugging the discarded segments back into the line updating the index of m in accordance with the received sequence. After reversing all partitioning steps the decoder knows the index of m within the original message set $\mathcal{M}$.

In the following we discuss in what way δ, p and k should be chosen such that the complexity of the algorithm is of order $\mathcal{O}(n)$. We need to make sure that the equations (6.10), (6.11) and (6.14) can be fulfilled. Those are the main limitations for our choices. According to those limitations our goal is to choose the parameters in a way that the complexities of encoder and decoder are minimized.

So first consider equation (6.10). By looking at equation (6.13) we find that equation (6.10) is fulfilled if

$$2^{-\delta\varepsilon_\tau} \leq \sqrt{\frac{\delta}{8\delta(1/2+\tau/2)\delta(1/2-\tau/2)}} = c_1(\tau)\sqrt{\frac{1}{\delta}}$$

for some $c_1(\tau)$ which is a constant depending on τ but independent of δ. This is equivalent to

$$2^{\delta\varepsilon_\tau} \geq \frac{\sqrt{\delta}}{c_1(\tau)} \tag{6.17}$$

and we know (see (6.12)) that there exists a constant $\delta_0(\tau, \varepsilon_\tau)$ independent of n such that inequality (6.17) holds for all $\delta \geq \delta_0(\tau, \varepsilon_\tau)$.

Next we consider inequality (6.14). This inequality is satisfied for $k \in \Theta\left(\frac{n}{\delta}\right)$. It is possible to choose $\delta \geq \delta_0(\tau, \varepsilon_\tau)$ to be a constant fulfilling (6.17). If this choice is made, it follows that $k \in \Theta(n)$ and $\delta k \in \Theta(n)$.

Finally we need to consider equation (6.11). Again by equation (6.13) we find that

equation (6.11) is fulfilled if

$$\begin{aligned}&\sqrt{\frac{\delta}{2\pi\delta(1/2+\tau/2)\delta(1/2-\tau/2)}}\\ &= c_2(\tau)\sqrt{\frac{1}{\delta}}\\ &\leq 2^{\delta k \frac{1+\tau}{2}\varepsilon_\tau},\end{aligned} \tag{6.18}$$

where some $c_2(\tau)$ depends only on τ. Since $k > \delta$ and because for their product it holds $\delta k \in \Theta(n)$ the inequality in (6.18) holds.

As $\delta \in \mathcal{O}_{\tau,\varepsilon_\tau}(1)$, the creation of the lookup tables for each partitioning step can be done in $\mathcal{O}_{\tau,\varepsilon_\tau}(1)$ time. Since at most $k \in \Theta(n)$ partitioning steps need to be performed, the entire partitioning strategy is of order $\mathcal{O}_{\tau,\varepsilon_\tau}(n)$. Furthermore, Weight Algorithm as well as Uncoded Algorithm are also of complexity order $\mathcal{O}(n)$. Therefore, the overall complexity of encoding and decoding algorithms is of order $\mathcal{O}_{\tau,\varepsilon_\tau}(n)$ which completes the proof. ■

6.4 Tightness of Upper and Lower Bound and Comparison to Previous Results

In this section we first graphically illustrate the obtained upper and lower bounds on $C_2^f(\Gamma_Z^2, \tau)$ specified in Theorems 6.1 and 6.2 in Fig. 6.4.1. For reference a lower bound obtained by a modification of the rubber method presented in [DLM21] is given.

The plot shows that the lower bound obtained in Theorem 6.2 (blue) is at least as large as the previously established lower bound in [DLM21] (orange). In fact the bounds are only equal for a finite number of points. Notice also that previous results only show that a positive rate can be achieved for $\tau < 1/2$, whereas the results presented in this chapter show that positive rates are achievable for any $\tau < 1$. The gap between upper (dashed red) and lower bound on $C_2^f(\Gamma_Z^2, \tau)$ is only small for small values of τ (and of course also for $\tau = 1$). Closer examination of Theorem 6.1 and Theorem 6.2 gives an approximation of $C_2^f(\Gamma_Z^2, \tau)$ for small τ.

Corollary 6.1. *For $\tau \to 0$, the capacity error function satisfies*

$$C_2^f(\Gamma_Z^2, \tau) = 1 + \tau \log_2 \tau + (1 - \log_2 e)\tau + o(\tau),$$

where e is Euler's number.

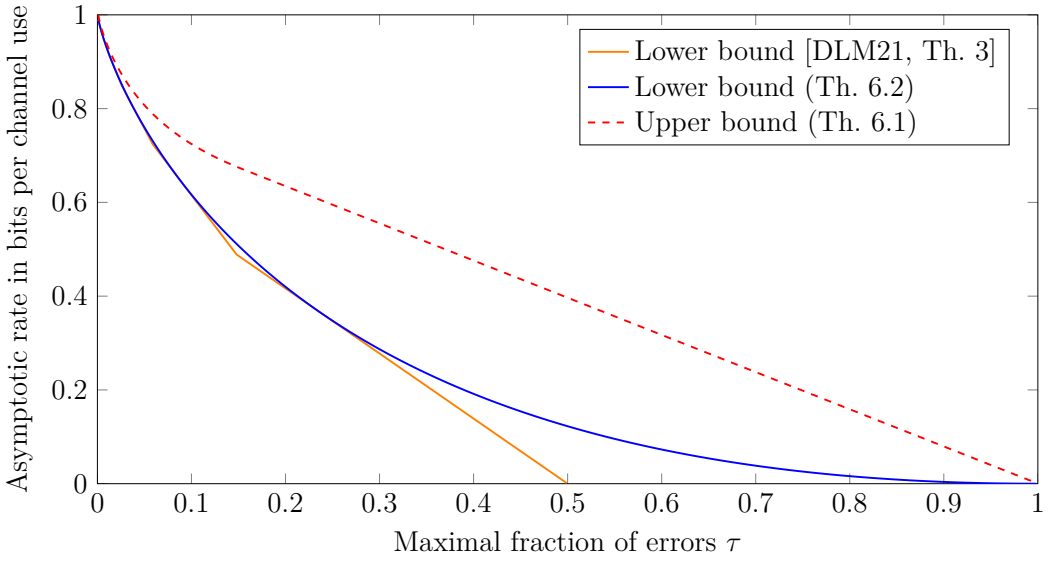

Figure 6.4.1: Bounds on the capacity error function for the Z-channel with noiseless feedback.

Proof. By Theorem 6.2, we obtain that for $\tau \to 0$

$$\begin{aligned} C_2^f(\Gamma_Z^2, \tau) &\geq (1+\tau) - (1+\tau)\log_2(1+\tau) + \tau \log_2 \tau \\ &= 1 + \tau - (1+\tau)\log_2(e)(\tau + o(\tau)) + \tau \log_2 \tau \\ &= 1 + \tau \log_2 \tau + (1 - \log_2(e))\tau + o(\tau) \ , \end{aligned}$$

where the first inequality can be derived by using the series expansion of the logarithm $\ln(1+x) = \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^k}{k}$.

By applying Theorem 6.1 and setting $\tau' := 0$ and $\omega := \omega(\tau) = \frac{1}{2} - \tau^{1/4}$, we get

$$C_2^f(\Gamma_Z^2, \tau) \leq \max\left\{1 - (\omega+\tau)h\left(\frac{\tau}{\omega+\tau}\right),\ h(\omega)\right\} \ .$$

Clearly, for $\tau \to 0$, it holds that

$$\begin{aligned} & 1 - (\omega+\tau)h\left(\frac{\tau}{\omega+\tau}\right) \\ = \ & 1 + \tau \log_2 \tau + \omega \log_2 \omega - (\omega+\tau)\log_2(\omega+\tau) \\ = \ & 1 + \tau \log_2 \tau + \omega \log_2 \omega \\ & -(\omega+\tau)\left(\log_2 \omega + \frac{\tau \log_2 e}{\omega} + o(\tau)\right) \\ = \ & 1 + \tau \log_2 \tau - \tau \log_2 \omega - \tau \log_2 e + o(\tau) \\ = \ & 1 + \tau \log_2 \tau + (1 - \log_2 e)\tau + o(\tau) \ . \end{aligned}$$

On the other hand, it is easy to verify that

$$\begin{aligned} & h(\omega) \\ = \; & h\left(\frac{1}{2}-\tau^{1/4}\right) \\ = \; & -\left(\frac{1}{2}-\tau^{1/4}\right)\log_2\left(\frac{1}{2}-\tau^{1/4}\right)-\left(\frac{1}{2}+\tau^{1/4}\right) \\ & \log_2\left(\frac{1}{2}+\tau^{1/4}\right) \\ = \; & 1-2\tau^{1/2}\log_2 e+o(\tau^{1/2}) \end{aligned}$$

again by using the series expansion of the natural logarithm. This completes the proof as $\tau^{1/2}$ dominates $\tau\log_2\tau$ in magnitude. ■

Remark 6.2. *Recall that (Theorem 5.1) the capacity error function of the symmetric channel with noiseless feedback is equal to* $1-h(\tau)=1+\tau\log_2\tau-\tau\log_2 e+o(\tau)$ *as* $\tau\to 0$. *Observe that this rate is smaller than the capacity error function correcting asymmetric errors by* $\tau+o(\tau)$.

We conjecture that the lower bound given in Theorem 6.2 is tight and justify this conjecture in the following. We recall the famous Hamming bound [MS77b] which limits the size $M_q(\Gamma_S^q,\tau)$ of a largest q-ary code of length n correcting t errors for the symmetric channel

$$M_q(\Gamma_S^q,\tau)\le A_q(n,t):=\frac{q^n}{\sum_{i=0}^{t}\binom{n}{i}(q-1)^i}\ . \tag{6.19}$$

It has been shown by Berlekamp in [Ber68] that this bound also holds for feedback codes. By [SY03; DS04], for a fixed number of errors t, $q=2$ and $n\to\infty$, the largest feedback code of length n correcting t asymmetric errors has size asymptotically $A_2(n+t,t)$. More specifically,

Lemma 6.2 (Follows from [SY03])**.** *Given a fixed integer* $t\ge 1$, *the maximum number of messages is asymptotically*

$$M_2(\Gamma_Z^2,n,t)=\frac{2^{n+t}t!}{n^t}(1+o(1))\quad \textit{as } n\to\infty\ . \tag{6.20}$$

It turns out that our lower bound is in spirit of this observation since it holds that

$$\underline{R}(\tau)=\lim_{n\to\infty}\frac{\log_2 A_2(n+\lceil\tau n\rceil,\lceil\tau n\rceil)}{n} \tag{6.21}$$

and therefore the construction we have presented in Section 6.3.1 achieves the Hamming bound for blocklength $n+\lceil\tau n\rceil$ correcting at most $\lceil\tau n\rceil$ asymmetric errors.

6.5 Summary and Open Problems

In this chapter, we discussed coding with noiseless feedback for the Z-channel in the combinatorial setting. By providing an explicit construction, we showed that the capacity error function $C_2^f(\Gamma_Z^2, \tau)$ is positive for any $\tau < 1$. We have shown encoding and decoding algorithms with complexity of order $\mathcal{O}(n)$ achieving our lower bound arbitrarily close. We constituted the conjecture that the lower bound on $C_2^f(\Gamma_Z^2, \tau)$ presented in Theorem 6.2 is tight for all τ. Furthermore, we presented an upper bound on $C_2^f(\Gamma_Z^2, \tau)$ that is close to the aforementioned lower bound for small values of τ. Determining $C_2^f(\Gamma_Z^2, \tau)$ remains an open research question for almost all values of τ. In accordance with our conjecture we especially consider investigating methods to construct better upper bounds a decent starting point for further research.

In this chapter, we considered feedback encoding for the case of noiseless instantaneous feedback within the encoding of every symbol. Considerations about limiting the utilization of the feedback may be an interesting starting point for subsequent research on the proposed error model. Another natural question to be asked is whether the Z-channel capacity (probabilistic setting) can be achieved by a similar encoding algorithm.

7
Unidirectional Channels

Abstract

In digital systems such as fiber optical communications the ratio between probability of errors of type $1 \to 0$ and $0 \to 1$ can be large. This behaviour can be approximated by assuming that only one type of errors can occur. These errors are called asymmetric. Unidirectional errors differ from asymmetric type of errors, here both $1 \to 0$ and $0 \to 1$ type of errors are possible, but in any submitted codeword all the errors are of the same type. We consider q-ary unidirectional channels with feedback and give lower and upper bounds on the capacity error function. It turns out that the bounds depend on the parity of the alphabet q. Furthermore we show that for feedback the capacity error function for the binary asymmetric channel is different from the symmetric channel. This is in contrast to the behavior of that function without feedback.

This chapter is based on our publication in Theoretical Computer Science [DLM21]. The author order has been decided to be alphabetical. The coauthors of this work are all senior researchers and major contributions to this work have been made by the author of this thesis. Parts of this work furthermore have been presented at the 2020 IEEE International Symposium on Information Theory (ISIT) [DML20].

7.1 Introduction and Motivation

Shannon analyzed in [Sha48] the capacity of a channel based on probabilistic notions. The arguably easiest example is a binary symmetric channel (BSC) with crossover probability ε. In [Sha56] it is shown that feedback does not increase the capacity of a BSC. In a probabilistic model the number of errors in a block of symbols (having different probability) can be any number between 0 and the blocklength n. In this work on the contrary, we analyze the situation when the fraction of erroneous symbols is upper bounded by $\tau = \frac{t}{n}$, where t denotes the maximal number of errors per block. Our focus in this chapter lies in the analysis of the capacity error function of unidirectional channels.

The analysis in this chapter is purely combinatorial. That means that we are not distinguishing between likely and unlikely errors. We only consider whether a certain

output may occur given that a certain symbol was chosen at the input of the channel.

The difference between symmetric, asymmetric and unidirectional errors is described in [Web89]. In a symmetric channel, any possible input symbol can be converted to every possible symbol contained in the output alphabet by the noise. On the contrary in an asymmetric channel there exists at least one input symbol that cannot be converted to the channel's entire output alphabet. Binary asymmetric channels model photon communication systems better than symmetric channels. Due to losses within the transmission medium, photons may not reach the receiver. However, the channel cannot generate photons on its own. Therefore the channel is of asymmetric nature [Web89, p. 1.3]. A unidirectional channel is composed of two special asymmetric channels. Before the transmission of each codeword one of them is selected. Within the transmission of the respective codeword symbols the channel behaves like the selected one. Before the transmission neither sender nor receiver know which channel has been selected. After the transmission of the codeword has been completed the channel is reselected before sending the next codeword. Unidirectional channels are formally specified in Definition 7.1.

The simplest asymmetric channel is the Z-channel with binary input and output alphabet (see also Chapter 6). For this channel a transmitted 0 symbol is always received correctly, whereas a transmitted 1 can be disturbed by the noise such that a 0 is received (Figure 7.3.1). For a transmission without feedback, the binary Z-channel has the same capacity error function as the symmetric channel (Figure 5.3.2). This has already been shown by Bassalygo in [Bas65]. The probabilistic model of the q-ary case was considered in [D'y75] and an upper bound on the error probability was given. Asymmetric and unidirectional channels without feedback are analyzed in numerous publications. The major results and references can be found in [MS77b] and [Web89]. In this chapter we give lower and upper bounds on the capacity error function of asymmetric and unidirectional channels in the presence of noiseless feedback. The lower bound shown in Section 7.6 is derived by generalizing the methods used in [DL19] and the lower bound presented in Section 7.7 is a generalization of the rubber method of [ADL05]. The lower bound using ideas of [DL19] works only for $q \geq 3$ whereas the lower bound using ideas of [ADL05] also works for $q = 2$. The proof of the upper bound uses the upper bound from [DL19], which bounds q-ary error-correcting codes with partial feedback and limited magnitude. For a wide range of τ, we show that the capacity error function for the unidirectional channel composed out of a Z-channel and an inverse Z-channel is larger in comparison to the symmetric channel for binary input and output alphabets. This is not the case for the same channels without feedback.

7.2 Chapter Outline

In Section 7.3 we introduce the channel models relevant in this chapter. Section 7.4 the zero error capacity of the previously introduced channel models is derived. Then in

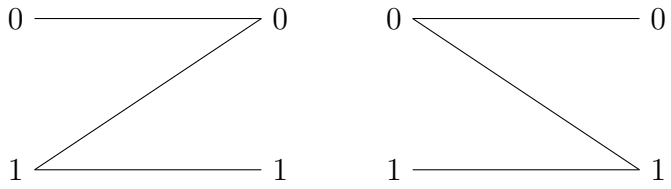

Figure 7.3.1: Binary Z-channel Γ_Z^2 and inverse Z-channel $\Gamma_{\overline{Z}}^2$

Section 7.5 an upper bound on the capacity error function of the discussed Z-channels and unidirectional channels is presented. For the construction of lower bounds two cases are distinguished. First in Section 7.6 a lower bound for alphabet sizes $q \geq 3$ is presented. For $q = 2$ it is shown in Section 7.7 it is shown how to adapt the rubber method for unidirectional channels. Finally, in Section 7.8 the results are summed up and some open problems are given.

7.3 Symmetric, Asymmetric and Unidirectional Errors

If the sender and the receiver share a channel with noiseless feedback, the capacity error function of the symmetric channel is only completely known for the binary case. The precise statement is given in Theorem 5.1.

In this chapter, we investigate channels for which the input alphabet $\mathcal{X}$ is equal to the channel's output alphabet $\mathcal{Y}$ and it holds that $\mathcal{X} = \mathcal{Y} = \mathcal{Q} := \{0, \ldots, q-1\}$, where q specifies the alphabet sizes. In the following we define the error vector $(e_1, e_2, \ldots, e_n)$ for the block transmission composed of n channel uses by

$$e_i := y_i - x_i \ , \tag{7.1}$$

where x_i and y_i denote the i-th channel input and channel output symbol, respectively and the subtraction denotes the usual subtraction over the real numbers.

Definition 7.1 (Unidirectional channel)**.** *A unidirectional channel is specified by two asymmetric channels having the same set of input symbols $\mathcal{X}$ and output symbols $\mathcal{Y}$. One of the channels allows only positive error vectors ($e_i \geq 0 \ \forall i$), whereas the other one only allows negative error vectors ($e_i \leq 0 \ \forall i$). Within each codeword the channel is specified by one of the asymmetric channel. Sender and receiver know both channels, but they do not know to which asymmetric channel the unidirectional channel corresponds. This may change for each codeword.*

In Figure 7.3.1 the binary unidirectional channel is shown. It is denoted as Γ_U^2 and is composed of the binary Z-channel and its counterpart, the inverse Z-channel.

Definition 7.2 (Generalized Z/inverse Z-channel)**.** *We specify the generalized Z-channel* Γ_Z^q *by the bipartite graph with input nodes*

$$\mathcal{X} = \mathcal{Y} = \mathcal{Q}$$

and the set of edges

$$\mathcal{E}_Z^q = \{(i, i-1) : i \in \{1, \ldots, q-1\}\} \cup \{(i, i) : i \in \{0, \ldots, q-1\}\}.$$

The inverse Z-channel $\Gamma_{\sum}^q$ *is specified by the bipartite graph with input nodes* $\mathcal{X} = \mathcal{Y} = \{0, \ldots, q-1\}$ *and the set of edges* $\mathcal{E}_{\sum}^q = \{(i, i+1) : i \in \{0, \ldots, q-2\} \cup \{(i, i) : i \in \{0, \ldots, q-1\}\}$.

Remark 7.1. *The q-ary Z-channel was introduced by Varshamov [Var73]. In this work we refer to this channel as the generalized Z-channel. It is a special case of an asymmetric channels where the errors are of limited magnitude, see [AAKT06] (here magnitude-1 errors). Codes designed for such channels can be applied to multilevel flash memories (see [CSBB10]).*

Remark 7.2. *The capacity error function of the generalized Z-channel* Γ_Z^q *is equal to the capacity error function of the inverse generalized Z-channel* $\Gamma_{\sum}^q$.

Proof. The inverse Z-channel $\Gamma_{\sum}^q$ can be obtained from the Z-channel Γ_Z^q by a bijective mapping. The labelling of the nodes at the input and the output is simply in reversed order. ■

Both channels are depicted in Figure 7.3.2. Combined they form a unidirectional channel which we denote as Γ_U^q.

The symbols 0 and $q-1$ are of special interest for the generalized Z-channel and the inverse generalized Z-channel. For the generalized Z-channel, the symbol 0 has the property that the sender knows that this symbol cannot be changed by the channel. However, if the symbol 0 is received it is possible that the sender has sent a 1 symbol. If the symbol $q-1$ is received, the receiver knows that the transmitter indeed sent this symbol. The properties of 0 and $q-1$ are swapped for the inverse generalized Z-channel compared to the generalized Z-channel. All other symbols do not have these special properties.

7.4 Zero-Error Capacity of Asymmetric and Unidirectional Channels

The method we use to obtain the zero error capacity of the generalized Z-channel was already introduced by Shannon in [Sha56, Theorem 7].

The idea is to split the input symbols into sets according to their connected outputs.

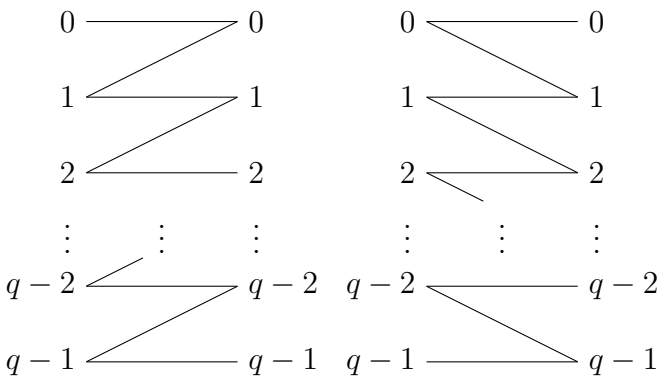

Figure 7.3.2: Generalized Z-channel and generalized inverse Z-channel

Theorem 7.1 (Thm. 7 [Sha56])**.** *We call two input letters adjacent if there is an output letter which can be caused by both input letters. If any two input letters are adjacent, the zero error capacity $C_{0,q}^f$ is zero. Let the set $\mathcal{S}_j$ denote the set of input symbols which are connected to the channel output $j \in \{0, \ldots, q-1\}$ and let P_i denote the probability of symbol i at the input of the channel. Let P_o be defined as*

$$P_o := \min_{P_i} \max_j \sum_{i \in \mathcal{S}_j} P_i \tag{7.2}$$

over all possible input distributions P_i.

If not every pair of input letters is adjacent, for the zero error capacity $C_{0,q}^f$ it holds that

$$C_{0,q}^f = \log_q \left(\frac{1}{P_o} \right) \quad . \tag{7.3}$$

Theorem 7.1 suggests that in order to find the zero error capacity, it is possible to find the input distribution which minimizes equation (7.2) and to take the logarithm of the reciprocal of the result.

Proposition 7.1. *The zero error capacity of the generalized Z-channel and the inverse generalized Z-channel is*

$$C_{0,q}^f(\Gamma_Z^q) = C_{0,q}^f(\Gamma_\Sigma^q) = \log_q \left(\left\lceil \frac{q}{2} \right\rceil \right) . \tag{7.4}$$

Proof. We apply Theorem 7.1 to the inverse generalized Z-channel Γ_Σ^q shown in Figure

7.3.2.

$$\begin{aligned}\mathcal{S}_0 &= \{0\}\\ \mathcal{S}_1 &= \{0,1\}\\ \mathcal{S}_2 &= \{1,2\}\\ &\vdots\\ \mathcal{S}_{q-1} &= \{q-2, q-1\}\end{aligned}$$

Consider now an arbitrary distribution on the set of input letters $\{0,\ldots,q-1\}$. It is obvious that

$$\sum_{i\in\mathcal{S}_0} P_i \leq \sum_{i\in\mathcal{S}_1} P_i \tag{7.5}$$

with equality if and only if $P_1 = 0$. Assuming that $P_1 > 0$, we get an input probability which is at least as good as before according to equation (7.2) if we change the input probability to

$$\begin{aligned}P'_0 &= P_0 + P_1\\ P'_1 &= 0\\ P'_2 &= P_2\\ &\vdots\\ P'_{q-1} &= P_{q-1}\,.\end{aligned}$$

This is because $\sum_{i\in\mathcal{S}_2} P_i = P_1 + P_2$ can only get smaller, while for $j \in \{3,\ldots,q-1\}, \sum_{i\in\mathcal{S}_j} P_i$ remains unchanged. Now we consider the distribution P'. As $P'_1 = 0$, the set $\mathcal{S}_2$ effectively only contains the element 2. As $\mathcal{S}_3 = \{2,3\}$ we can use the same method as before to show that the zero error capacity cannot get smaller if we change the distribution such that $P''_2 = P'_2 + P'_3$ and $P'_3 = 0$ while all other input probabilities remain the same.

We use this argument now inductively. Therefore, it is optimal to have only non-zero probability for every second input symbol. Looking at equation (7.2) we see that the optimal input distribution is the uniform distribution over all input symbols k fulfilling $k \equiv 0 \pmod 2$. Therefore, the zero error capacity of the inverse generalized Z-channel is

$$C^f_{0,q}(\Gamma^q_\Sigma) = \log_q\left(\left\lceil\frac{q}{2}\right\rceil\right)\,. \tag{7.6}$$

Remark 7.2 implies that the zero error capacities of the generalized Z-channel and the inverse generalized Z-channel are equal. ■

Remark 7.3. *Note also that by precisely the same argument (as for Proposition 7.1, we have the following more general fact: For a q-ary Z-channel (for each transmitted symbol i the output $y(i) \leq i$) where the errors are of magnitude l, the zero error capacity $C_{0,q}^{f}(\Gamma_Z, l) = \log_q \left\lceil \frac{q}{l+1} \right\rceil$. Clearly the same holds for the inverse channel ($y(i) \geq i$) and for the unidirectional channel we have $C_{0,q}^{f}(\Gamma_U, l) \leq \log_q \left\lceil \frac{q}{l+1} \right\rceil$.*

Furthermore, Proposition 7.1 can be used to show the following.

Proposition 7.2. *It holds $C_{0,q}^{f}(\Gamma_U^q) = \log_q \left(\left\lceil \frac{q}{2} \right\rceil \right)$.*

Proof. Because of Property 7.1, it holds that $C_{0,q}^{f}(\Gamma_Z^q) = C_{0,q}^{f}(\Gamma_{\sum}^q) = \log_q \left(\left\lceil \frac{q}{2} \right\rceil \right)$. Therefore, $\mathcal{C}_{0,q}^{f}(\Gamma_U^q) \leq \log_q \left(\left\lceil \frac{q}{2} \right\rceil \right)$. It remains to be shown that $C_{0,q}^{f}(\Gamma_U^q) \geq \log_q \left(\left\lceil \frac{q}{2} \right\rceil \right)$. In order to prove this, we propose the following encoding and strategies.

For this we define the set $\mathcal{A} := \{k \in \{0, \ldots, q-1\} : k \equiv 0 \pmod 2\}$.

The encoding function

$$c : \mathcal{M} \to \mathcal{A}^{n-1} \tag{7.7}$$

can be any bijective function between $\mathcal{M}$ and $\mathcal{A}^{n-1}$. If the sender would like to send message $m \in \mathcal{M}$, he computes $x^{n-1} = c(m)$ and sends it over the channel. In order to send the last symbol, the sender distinguishes two cases:

1. The channel caused an error within at least one of the previously transmitted symbols. In this case the sender knows the channel and sends $x_n = 0$ if the active channel is Γ_Z^q and $q-1$ otherwise.

2. No error has occurred within the transmission. In this case the sender sends $x_n = 0$.

We denote the vector of symbols at the channel output as $\mathbf{y}^n$. In order to determine which channel was active during the transmission, the decoder checks the value of y_n. There are three cases to be distinguished:

1. $y_n = 0$: Either no transmission error occurred at all or the active channel was Γ_Z^q.

2. $y_n = q-1$: The active channel was $\Gamma_{\sum}^q$.

3. $y_n = 1$: No error occurred within the first $n-1$ symbols and the active channel was $\Gamma_{\sum}^q$.

Let $\mathcal{K}_i$ denote the set of possible outputs for the channel if the symbol i is transmitted. The sets $\mathcal{K}_i$ for different inputs are disjoint for Γ_Z^q and $\Gamma_{\sum}^q$, respectively. From the last symbol y_n, the decoder knows which channel has been active. Therefore our coding strategy is successful for a message set $\mathcal{M}$ of size

$$|\mathcal{M}| = \left\lceil \frac{q}{2} \right\rceil^{n-1}$$

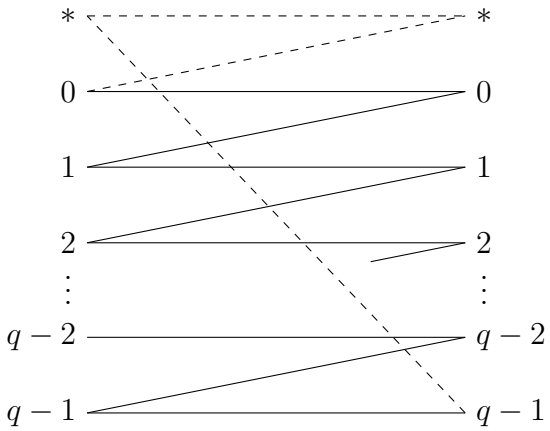

Figure 7.5.1: Bipartite graph Γ^{q*}

achieving asymptotically the rate

$$R := \log_q \left(\left\lceil \frac{q}{2} \right\rceil \right).$$

■

Remark 7.4. *It was shown in [AAKT06] that for the zero error capacity of the unidirectional channel (without feedback) with error magnitude* $0 \leq l \leq q-1$ *it holds that*

$$C_{0,q}(\Gamma_U, l) = \log_q \left\lceil \frac{q}{l+1} \right\rceil.$$

7.5 An Upper Bound on Capacity Error Functions

The analysis in the previous section deals with the zero error capacity, assuming that the number of errors in a block of length n can be from 0 to n. This section deals with the problem of allowing the channel to only change a fraction of the symbols during the transmission. Let this fraction be denoted as $\tau = \frac{t}{n}$, where t denotes the number of possible errors.

An upper bound on the capacity error function of the Z-channel can be obtained by using an upper bound on the maximal set of messages, which can successfully be transmitted over the channel Γ^{q*} depicted in Figure 7.5.1.

Definition 7.3. *The channel* Γ^{q*} *is specified by the graph with nodes* $\mathcal{V}_{in} = \mathcal{V}_{out} = \{*\} \cup \{0, \ldots, q-1\}$ *and the set of edges* $\mathcal{E} = \{(*,*), (i,i) : i \in \{0, \ldots, q-1\}\} \cup \{(i, i-1) : i \in \{1, \ldots, q-1\}\} \cup \{(*, q-1), (0, *)\}\}$.

Compared to the channel Γ_Z^q the alphabets at the input and the output of the channel Γ^{q*} are enlarged by the symbol $*$. We use this channel to bound the capactiy

error function $C_q^f(\Gamma_U^q, \tau)$. It is possible to create the channel Γ_Z^q from Γ^{q*} by not using the $*$ symbol and by mapping each $*$ symbol at the output to the 0 symbol at the receiver.

Let $\mathcal{M}_q^Z(n,t)$ denote a maximal set of messages which can successfully be transmitted over Γ_Z^q, let $\mathcal{M}_q^U(n,t)$ denote a maximal set of messages which can successfully be transmitted over Γ_U^q and let $\mathcal{M}_{q*}(n,t)$ denote a maximal set of messages which can successfully be transmitted over Γ^{q*}.

Theorem 7.2. *Let*

$$R_q^{upper}(\tau) := 1 + h_q\left(\min\left(\tau, \frac{1}{q+1}\right)\right) - \min\left(\tau, \frac{1}{q+1}\right) - h_q\left(\min\left(\tau, \frac{1}{2}\right)\right) \ .$$

Then it holds that

$$C_q^f(\Gamma_U^q, \tau) \leq C_q^f(\Gamma_Z^q, \tau) \leq R_q^{upper}(\tau) \ , \tag{7.8}$$

where h_q denotes the binary entropy function with logarithms to the base q.

Proof. We will show that the following inequalities hold:

$$\mathcal{M}_q^U(n,t) \overset{(a)}{\leq} \mathcal{M}_q^Z(n,t) \overset{(b)}{\leq} \mathcal{M}_q^{upper}(n,t) \ , \tag{7.9}$$

where

$$\mathcal{M}_q^{upper}(n,t) := \frac{\sum_{i=0}^t \binom{n}{i} q^{n-i}}{\sum_{i=0}^t \binom{n}{i}} \ . \tag{7.10}$$

(a) follows because every successful algorithm for Γ_U^q is also a successful algorithm for Γ_Z^q. In order to show (b), we first note that every successful strategy on Γ_Z^q is also successful on Γ^{q*} by not using the symbol * at the input and interpreting each * at the output as 0. At the output of Γ^{q*}, there are at most t $*$-symbols using such an encoding strategy. In order to compute the cardinality of the set of output sequences $\mathcal{S}$, they are partitioned into the sets $\mathcal{S}_0, \ldots, \mathcal{S}_t$, where $\mathcal{S}_i$ denotes the set of output sequences containing $*$ exactly i times. The cardinality of these sets is

$$|\mathcal{S}_i| = \binom{n}{i} q^{n-i} \ . \tag{7.11}$$

Equation (7.10) results from the fact that the $\mathcal{S}_i$ are disjoint and a sphere packing argument.

Next we show that $\lim_{n\to\infty} \frac{M_q^{upper}(n,t)}{n} = C_q^{upper}(\tau)$.

This limit is determined by the largest terms in both sums of equation (7.10).

The largest term of the sum in the denominator is equal to $\binom{n}{\min(t,n/2)}$. In order to deal with the sum in the numerator, we use Stirling's approximation to estimate the

binomial coefficients of

$$\lim_{n\to\infty} \frac{\log_q\left(\sum_{i=0}^{t}\binom{n}{i}q^{n-i}\right)}{n} \ ,$$

neglecting terms which are asymptotically irrelevant. The sum in the numerator omitting all asymptotically irrelevant terms and using Stirling's approximation (see [Kra+08]) is given by

$$q^n \sum_{i=0}^{t} q^{n(h_q(\xi_i)-\xi_i)} \leq q^n(\tau n+1) \max_{\xi_i\in[0,\tau]} q^{n(h_q(\xi_i)-\xi_i)} \ , \tag{7.12}$$

where we defined $i := \xi_i n$ and $\tau := \frac{t}{n}$.

Using elementary differential calculus, one obtains that the exponent is maximized for $\xi_i = \min\left(\tau, \frac{1}{q+1}\right)$.

Therefore,

$$\lim_{n\to\infty} \frac{\log_q(\sum_{i=0}^{t}\binom{n}{i}q^{n-i})}{n} = 1 + h_q\left(\min\left(\tau, \frac{1}{q+1}\right)\right) - \min\left(\tau, \frac{1}{q+1}\right) \ . \tag{7.13}$$

As n goes to infinity, the error due to Stirling's approximation vanishes and using similar arguments for the sum in the denominator, we get

$$\begin{aligned}&\lim_{n\to\infty} \frac{\log_q(\mathcal{M}_q^{upper}(n,t))}{n}\\ &= 1 + h_q\left(\min\left(\tau, \frac{1}{q+1}\right)\right) - \min\left(\tau, \frac{1}{q+1}\right) - h_q\left(\min\left(\tau, \frac{1}{2}\right)\right) \ ,\end{aligned}$$

completing the proof. ■

Remark 7.5. *In the theorem, we follow the idea of Hamming to get an upper bound. The new idea we use is to extend the channel without using all symbols at the input. Therefore we reduce the number of output sequences, obtaining a better upper bound on the number of messages using the sphere packing argument. To the best of our knowledge, this method is new. Perhaps it can be used for other problems to get new upper bounds.*

7.6 Lower Bound on the Capacity Error Function

In this section a method introduced in [DL19, Lemma 2] is recapitulated. It is then used to construct a lower bound on the capacity error function of Γ_Z^q and Γ_Σ^q. In the following, the set of graphs with vertices $\mathcal{V}_{in} = \mathcal{V}_{out} = \{0,\ldots,q-1\}$ all having at most degree 2 is denoted as $\tilde{\Gamma}_2(q)$.

Lemma 7.1 ([**DL19**], Lemma 1)**.** *Let $R > 0$ be the rate of a successful algorithm for a channel $\Delta \in \tilde{\Gamma}_2(q)$ and $\tau = 1$. Then there exists a successful algorithm with rate $1 - \log_q(2)$ for all τ.*

Lemma 7.2 ([**DL19**], Lemma 2)**.** *If there exists a successful algorithm for a channel $\Delta \in \tilde{\Gamma}_2(q)$ with rate $R > 0$ for all τ, then there exists a successful algorithm with rate $1 - h(\tau)\log_q(2)$ for Δ and $0 \leq \tau \leq \frac{1}{2}$, where $h(\cdot)$ denotes the binary entropy function.*

We omit the proof of Lemma 7.1 and give only the idea of the construction achieving the rate proposed in Lemma 7.2. A full proof is given in [DL19].

Proof sketch: The strategy of the proof of [DL19] is to give an algorithm which can transmit $\frac{q^n}{\binom{n}{t}}$ messages. This algorithm then achieves the rate claimed in the Lemma. The algorithm uses the first $k_1 = n - \log_q \binom{n}{t}$ symbols to transmit the information symbols and to use the rest of the block to correct potential errors caused by the channel. It is sufficient for the receiver to know the error positions because the nodes at the receiving side of the graph only have a degree of at most 2. Let t_1 denote the number of errors within the information symbols and $n_1 := n - k_1 = \log_q \binom{n}{t}$.

If $\binom{k_1}{t_1} \leq q^{Rn_1}$, then the successful algorithm for $\tau = 1$ can be used to transmit the error positions, and since this works for any number of errors, the receiver can decode the message correctly. Otherwise, let $k_2 = \log_q \binom{k_1}{t_1}$ and use the interval $[k_1 + 1, k_1 + k_2]$ to send the error positions of the t_1 errors within the first interval. This process continues by defining $k_{i+1} = \log_q \binom{k_i}{t_i}$ and $n_{i+1} = n_i - k_{i+1}$ and is stopped when $\binom{k_i}{t_i} \leq q^{Rn_i}$. Then the successful algorithm for $\tau = 1$ can be used to transmit the error positions with certainty. In the last block, the k_i have to be transmitted as well. The amount of symbols that need to be reserved in order to do this is of order $\mathcal{O}(\log(n)\log(t))$. This does not change the asymptotically achievable rate of the algorithm. ■

The capacity error function for the Z-channel Γ_Z^q with an additional edge from input 0 to output $q-1$ was analyzed by Deppe and Lebedev in [DL19]. We denote this channel by Γ_{DL}^q and its error capacity error function as $C_q^f(\Gamma_{DL}^q, \tau)$. Deppe and Lebedev have shown the capacity error function of this channel for $q > 3$.

Theorem 7.3 ([DL19], Thm. 1)**.** *Let*

$$R_{DL}^q(\tau) := \begin{cases} 1 - h(\tau)\log_q 2 & \text{if } 0 \leq \tau \leq \frac{1}{2} \\ 1 - \log_q 2 & \text{if } \tau > \frac{1}{2} \ . \end{cases} \tag{7.14}$$

Then for $q > 3$ it holds that

$$R_{DL}^q(\tau) = C_q^f(\Gamma_{DL}^q, \tau) \tag{7.15}$$

and for $q = 3$ that

$$R_{DL}^q(\tau) \geq C_q^f(\Gamma_{DL}^q, \tau) \ . \tag{7.16}$$

Remark 7.6. *The problem of determining $C_3^f(\Gamma_{DL}^3, \tau)$ is still open. The reason why the achievability part of the proof in [DL19] is not working for $q = 3$ is that an algorithm with positive rate is required for $\tau = 1$, i.e., $C_{0,3}^f(\Gamma_{DL}^3) > 0$ is required, to make Lemma 7.1 and Lemma 7.2 applicable. This is only the case if $q > 3$.*

Since the additional edge can only make the channel worse, we have

$$C_q^f(\Gamma_Z^q, \tau) \geq C_q^f(\Gamma_{DL}^q, \tau) \ . \tag{7.17}$$

Corollary 7.1. *Let $q \geq 3$. Then the capacity error functions of Γ_Z^q, Γ_Σ^q and Γ_U^q are lower bounded by R_{DL}^q.*

$$C_q^f(\Gamma_Z^q, \tau) \geq C_q^f(\Gamma_U^q, \tau) \geq R_{DL}^q \tag{7.18}$$

Proof. The algorithm in the proof of Lemma 7.2 requires the possibility of error-free transmission at a positive rate for $\tau = 1$. The reason for this is outlined in the proof sketch of Lemma 7.2. While this is not possible for Γ_{DL}^3 this is possible for Γ_U^3 for $\tau = 1$ by Proposition 7.2. The remaining steps to prove Corollary 7.1 follow analogously to the proof of Theorem 7.3. ■

We note that Corollary 7.1 does not give information on the capacity error functions for the binary channels.

Corollary 7.2. *For odd q the following equations hold for $\tau \geq \frac{1}{2}$:*

$$C_q^f(\Gamma_Z^q, \tau) = C_q^f(\Gamma_U^q, \tau) = C_{0,q}^f(\Gamma_Z^q) = C_{0,q}^f(\Gamma_U^q) \tag{7.19}$$

Proof. In accordance with the definition in the proof of Theorem 7.2, it holds that $R_q^{upper}(\tau) = \log_q\left(\frac{q+1}{2}\right)$ for $\tau \geq \frac{1}{2}$, which is equal to $\log_q(\lceil \frac{q}{2} \rceil) = C_{0,q}^f(\Gamma_U^q)$ for odd q by Proposition 7.1. ■

Figure 7.6.1 illustrates the results obtained in Theorem 7.2, Property 7.1 and Corollaries 7.1 and 7.2. Property 7.1 shows that the methodology proposed in [DL19] (Lemma 7.1 in this work) only achieves the zero error capacity of Γ_Z^q if q is even. However, the strategy proposed in the proof of Proposition 7.2 is able to achieve the zero error capacity for Γ_U^q and Γ_Z^q for any q. The results in Figure 7.6.1 are in accordance with Corollary 7.2 for $q = 5$. For even q we only know the zero error capacity which is achieved by the lower bound. For $\tau \neq 1$ our results do not show whether upper and lower bounds on $C_q^f(\Gamma_U^q, \tau)$ are tight. $q = 6$ was chosen as an example for even q.

7.7 Constructing Lower Bounds Using Rubber Methods

The rubber method was developed in [ADL05] and generalized in [Leb16]. It is used to determine a lower bound on the capacity error function of a channel. Basically

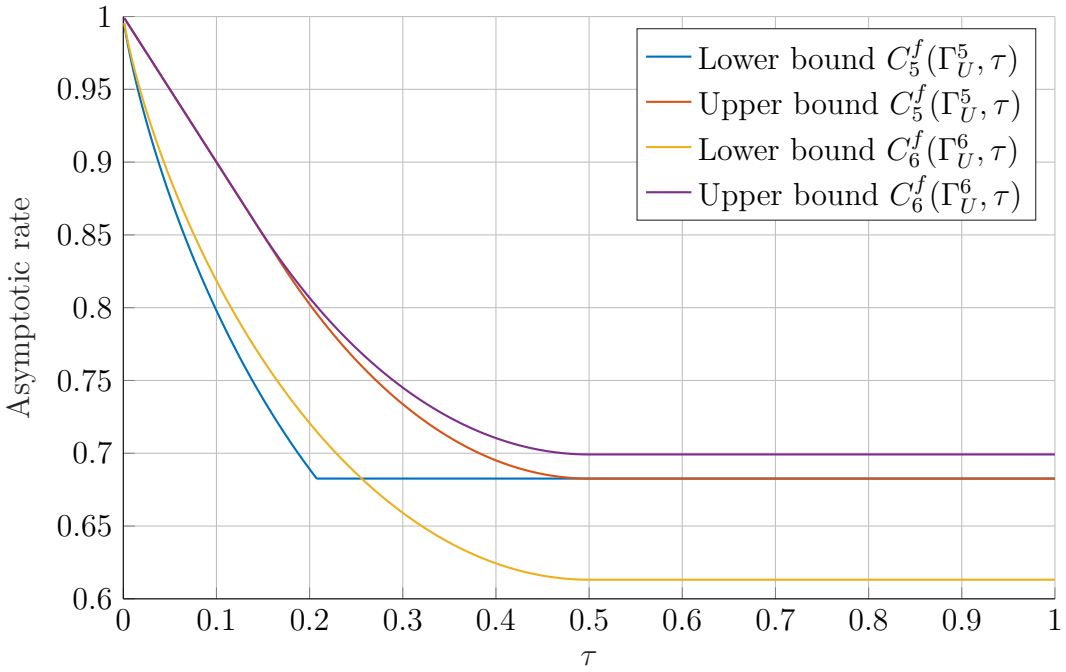

Figure 7.6.1: Lower and upper bound on the capacity error function of the unidirectional channels Γ_U^5 and Γ_U^6

it reserves a sequence of symbols to notify the receiver that his previously received symbol has been an erroneous one. Therefore, after receiving the rubber sequence, the receiver deletes the rubber plus the previous symbol. In [ADL05], a rubber sequence consists of r equal rubber symbols $b \in \mathcal{Q}$. The sender then retransmits the previously erroneous symbol again. Because the number of errors t is fixed and smaller than the blocklength n, he will succeed after at most t trials. In [ADL05] it is shown that that the rubber method is a successful algorithm. This means the decoder is able to decode each message correctly. In the error analysis in [ADL05], the authors consider two kinds of errors: a **standard error** (which means a symbol is changed to another symbol and the sender sends the rubber sequence) and a **towards rubber error** (which means a symbol is changed to a rubber symbol such that the receiver obtains a rubber sequence). If a **towards rubber error** occurs, a correctly received symbol is deleted and has to be sent again. For the generalized Z-channel Γ_Z^q, a **towards rubber error** is not possible if we use r times $q-1$ as the rubber sequence. Also, for the inverse generalized Z-channel Γ_{Σ}^q, a **towards rubber error** is not possible if we use r times 0 as the rubber sequence. Therefore, the sender does not have to retransmit the previously erroneous symbol again, because for those channels there is only the possibility of a **standard error** and in case the decoder knows the correct symbol since at most two different input symbols can be mapped to a specific output symbol by the channel. We denote this modified algorithm (without retransmissions)

by $A(r,b)$ if we use $b^r = (b,\ldots,b)$ as the rubber and we get:

Lemma 7.3. *The modified rubber strategy $A(r,q-1)$ [$A(r,0)$] is a successful algorithm for the the generalized [inverse] Z-channel Γ_Z^q [Γ_∇^q].*

In the following, we denote this rubber method without retransmission as the modified rubber method.

To calculate the rate of this algorithm we need, as in [ADL05], the following definitions. Let $z_r^{r+1} = qz_r^r - q + 1$. It is well known that for $n \to \infty$ the number of sequences of length n not containing a block $b^r = (b,\ldots,b)$ is asymptotically equal to z_r^n (see [ADL05; Bru77], how to choose the initial value for the iteration). Furthermore, the following Lemma has been shown in [ADL05].

Lemma 7.4 ([ADL05], Thm. 2)**.** *For the r-rubber method (including retransmissions) it holds that the rate function is a tangent to $H_q(\tau)$ going through the point $\left(\frac{1}{r+1}, 0\right)$, where $H_q(\tau) := 1 + h(\tau)\log_q(2) - \tau\log_q(q-1)$.*

This lemma gives an alternative to obtaining z_r as one can just compute the tangent to $H_q(\tau)$ through the specified point solving the resulting equations for z_r.

Theorem 7.4. *Let z_r be defined as above. Then for the generalized [inverse] Z-channel Γ_Z^q [Γ_∇^q] for $q \geq 2$ we get*

$$C_q^f(\Gamma_Z^q, \tau) \geq R_{mr} := \begin{cases} \max\limits_{2\leq r\in\mathbb{N}} (1-r\tau)\log_q z_r & \text{if } 0 \leq \tau \leq \frac{1}{2} \\ 0 & \text{if } \tau > \frac{1}{2}\,. \end{cases}$$

Proof. The modified r-rubber method can be used to transmit information vectors not containing the rubber sequence. The blocklength is n and the number of erroneous symbols is at most t. For each error we require r symbols for correction. Therefore, the length of the information vectors can be at most $n - rt$. The number of these sequences is asymptotically equal to z_r^{n-rt}. Therefore the asymptotic rate achieved by the modified r-rubber method is

$$(1-r\tau)\log_q z_r.$$

This gives the following lower bound on the capacity error function

$$C_q^f(\Gamma_Z^q, \tau) \geq (1-r\tau)\log_q z_r.$$

We use the modified rubber method with the rubber length r that achieves the highest asymptotic for the respective value of τ and obtain the lower bound R_{mr}. ■

Remark 7.7. *For the generalized Z-channel Γ_Z^q, the modified rubber method with a single 0 as the rubber symbol achieves the rate*

$$R = (1-\tau)\log_q(q-1)\,. \tag{7.20}$$

Proof. Using Lemma 7.3, it is easy to see that for every erroneous symbol one extra symbol has to be transmitted in order to achieve error free transmission. Therefore it is possible to achieve a message set $\mathcal{M}$ of cardinality

$$M = (q-1)^{n-t}\,, \tag{7.21}$$

leading to a rate

$$R = (1-\tau)\log_q(q-1)\,. \tag{7.22}$$

■

By an adjustment of the modified rubber method for the channels Γ_Z^q and Γ_Σ^q we obtain the following result:

Theorem 7.5. *Let Γ_U^q be a unidirectional channel consisting of the generalized Z-channel Γ_Z^q and the inverse generalized Z-channel Γ_Σ^q ($q \geq 2$). Then we have*

$$C_q^f(\Gamma_U^q, \tau) \geq R_{mr}.$$

Proof. We adapt the modified rubber method to make it usable for Γ_U^q. Furthermore, we show that this method is successful and achieves the same asymptotic rate as the modified rubber method. The sender starts by using the encoding strategy for the generalized Z-channel Γ_Z^q and then adapts if he discovers that the active channel is Γ_Σ^q. After all the data symbols are transmitted, an extra symbol (either 0 or $q-1$ according to the channel) is added to tell the receiver which channel was present. The receiver can thus adjust his decoding algorithm accordingly. The previously described steps are now elaborated in more detail.

The only additional difficulty compared to the situation in Theorem 7.3 is that it is not known to the sender and the receiver whether Γ_Z^q or Γ_Σ^q is used for the transmission of each codeword but once the first error occurs the sender knows which channel is in action.

At the start of the transmission, the sender assumes that Γ_Z^q is the active channel of Γ_U^q. He therefore uses the modified rubber method for Γ_Z^q of length r as his encoding strategy.

When the channel inflicts the first error to the transmission, the sender knows which channel is active. We denote the transmitted symbol corresponding to the first error with x_{fe}. According to the actual channel, the sender adjusts his encoding strategy henceforth. We distinguish two cases:

1. The active channel is Γ_Z^q. In this case the channel keeps its encoding strategy as it is. After the sender knows that the all information symbols can be correctly retrieved by the decoder, he fills the remaining block with 0 symbols. Notice that this last step involves knowledge about the decoding algorithm at the sender.

2. The active channel is Γ_Σ^q. In this case there are two cases to be distinguished

a) A **standard error** occurred: In this case the transmitter sends r times the symbol $q-1$. If there are not enough symbols left in the block, the sender sends $q-1$ for the remaining symbols.

b) a **towards rubber error** occurred: In this case the transmitter sends $r-1$ times the symbol $q-1$. If there are not enough symbols left in the block, the sender sends $q-1$ for the remaining symbols.

The modified rubber method for Γ_{Σ}^{q} is used from now on to retransmit x_{fe} and to send the remaining information symbols. But before sending them, they are adapted to the channel Γ_{Σ}^{q} by using the mapping

$$\begin{aligned} f &: \mathcal{Q} \to \mathcal{Q} \\ k &\mapsto k+1 \pmod{q}. \end{aligned}$$

Potential remaining symbols within the block are filled by sending $q-1$ symbols.

Let the received sequence be denoted as $y^n = (y_1, \ldots, y_n)$. The decoding strategy depends on the symbol y_n:

1. $y_n = 0$: The decoder uses the decoding procedure for the modified rubber method for Γ_{Z}^{q} until he retrieves the message m. Potential remaining symbols within the block are ignored by the decoder.

2. $y_n = q-1$: The decoder checks for the first occurrence of the sequence r times $q-1$. All previously sent symbols, with exception of the symbol that was sent right before this sequence, have been received correctly. If those information symbols are sufficient to obtain the message with certainty, the decoder outputs the message m. Otherwise, the receiver continues its decoding procedure by using the decoding algorithm of the modified rubber method for Γ_{Σ}^{q} on the remaining symbols. The function

$$\begin{aligned} f^{-1} &: \mathcal{Q} \to \mathcal{Q} \\ k &\mapsto k-1 \pmod{q} \end{aligned}$$

is then to be applied on all information symbols which were sent after the first occurrence of the sequence r times $q-1$ to obtain the information vector. This vector uniquely determines the message m.

If the first error occurs within the last r symbols, this error cannot be corrected anymore, but the receiver knows that all previous symbols have been received correctly. Those contain the information about the message, and thus this case is not problematic for successful decoding. ■

The following result follows directly from Remark 7.7 and Theorem 7.5.

Corollary 7.3. *The asymptotic rate achieved by using the modified rubber method on the unidirectional channel Γ_U^q using a single symbol rubber is*

$$R = (1-\tau)\log_q(q-1)\,. \tag{7.23}$$

Remark 7.8. *The values for z_r in Theorem 7.4 and Theorem 7.5 can be computed. An important case is $q = 2$. It was proved that*

$$C_2^f(\Gamma_S^2, \tau) = (1-3\tau)\log_2\left(\frac{1+\sqrt{5}}{2}\right) \tag{7.24}$$

for $(3+\sqrt{5})^{-1} \leq \tau \leq 1/3$ (see Theorem 5.1). In [ADL05] it was shown that this capacity error function can be achieved with the rubber method.

If we apply the modified rubber method to the binary case we get the following result. The capacity error function of the unidirectional channel is lower bounded by

$$C_2^f(\Gamma_U^2, \tau) \geq \begin{cases} \max\limits_{2\leq r\in\mathbb{N}} (1-r\tau)\log_2 z_r & \text{if } 0 \leq \tau \leq \frac{1}{2} \\ 0 & \text{if } \tau > \frac{1}{2}\,. \end{cases}$$

The result for $r = 2$ is

$$(1-2\tau)\log_2\left(\frac{1+\sqrt{5}}{2}\right).$$

The major change compared to the capacity error function for the symmetric channel shown in Theorem 5.1 is the change of the factor $(1-3\tau)$ to $(1-2\tau)$. It occurs because retransmissions after erroneous symbols are unnecessary. Figure 7.7.1 shows the lower bound on $C_2^f(\Gamma_U^2, \tau)$ obtained by using the modified rubber method and $C_2^f(\Gamma_S^2, \tau)$ for comparison. This result is different without feedback, since the capacity error functions of symmetric and unidirectional channels are the same in that case.

7.8 Summary and Open Problems

In this work we have analyzed channels with feedback with a fixed maximal fraction of erroneous symbols. A major focus of this work has been asymmetric and unidirectional channels. Notably, we have shown that the capacity error function with feedback for the unidirectional channel composed of Z-channel and inverse Z-channel is larger than the capacity error function of the symmetric channel for a significant proportion of the possible values of τ. This is not the case for the same channels without feedback.

The lower bounds on the capacity error function that we derived in Section 7.6 and Section 7.7 are higher than the capacity error function for the channels for the non-feedback case. The lower bound in Section 7.6 only works for $q \geq 3$. In these cases it performs better than the lower bound in Section 7.7. However, the lower bound in

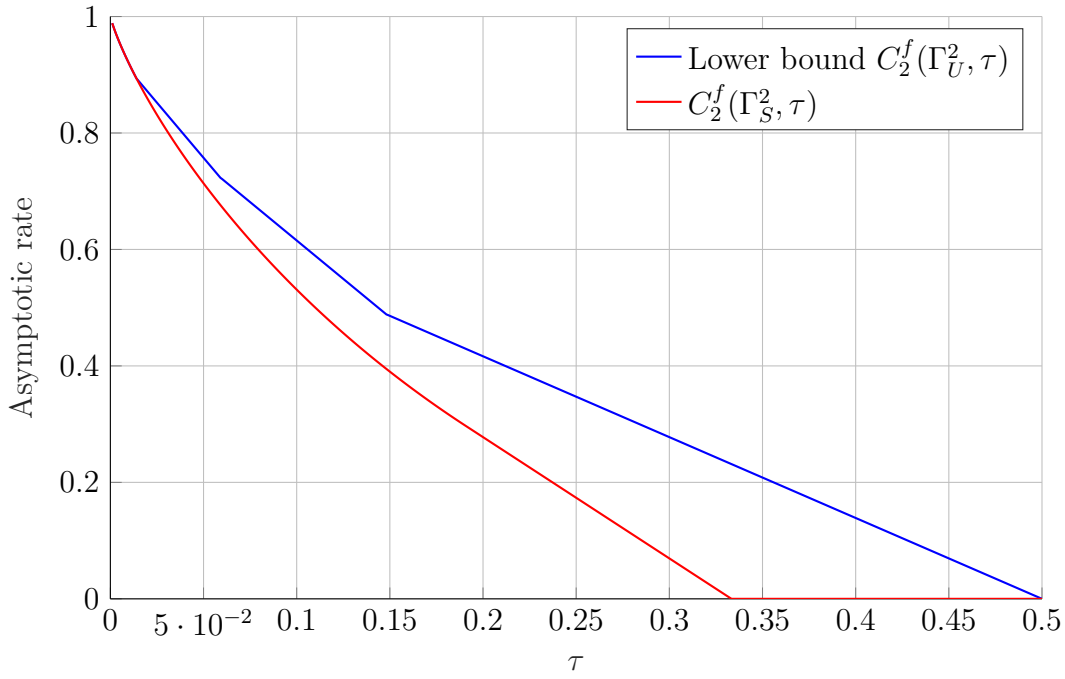

Figure 7.7.1: Comparison of the lower bound on $C_2^f(\Gamma_U^2, \tau)$ obtained by using the modified rubber method and the capacity error function of the symmetric channel $C_2^f(\Gamma_S^2, \tau)$

Section 7.7 works also for $q = 2$. Furthermore, the implementation of the encoding and decoding algorithm of this strategy is simpler.

The channels analyzed in this chapter have at most two input symbols connected to each output. If the channel model is known to the receiver, knowledge about an erroneous position at the receiver is sufficient to also reconstruct the transmitted symbol, hence additionally communicating information about the originally transmitted value is unnecessary. This can be used to create encoding strategies achieving a higher rate for many channels using the modified rubber method. Furthermore, it has been shown how to change the modified rubber method for unidirectional channels. The method proposed shows that it is possible to achieve the same rate for the unidirectional channel consisting of the generalized Z-channel and the inverse generalized Z-channel as for its components when the rubber method is used. Obtaining tighter bounds on the capacity error functions for several of the channels analyzed in this chapter is an interesting topic for further research.

8

Insertion-Deletion Channel

Abstract

In this chapter, a new problem of transmitting information over the adversarial insertion-deletion channel with feedback is introduced. Suppose that the encoder transmits n binary symbols one-by-one over a channel, in which some symbols can be deleted and some additional symbols can be inserted. After each transmission, the encoder is notified about the insertions or deletions that have occurred within the previous transmission and the encoding strategy can be adapted accordingly. The goal is to design an encoder that is able to transmit error-free as much information as possible under the assumption that the total number of deletions and insertions is limited by $\lceil \tau n \rceil$, $0 < \tau < 1$. We show how this problem can be reduced to the problem of transmitting messages over the substitution channel. Thereby, the capacity error function of the insertion-deletion channel with noiseless feedback is completely established.

This chapter is based on [MPVW21b] published in Problems of Information Transmission. Parts of this work furthermore have been presented at the 2020 Information Theory Workshop (ITW) [MPVW21a].

8.1 Introduction

The first important result in the construction of codes capable of correcting insertion and deletion errors dates back to 1965, when Levenshtein [Lev66] established that any code capable of correcting t deletions is capable of correcting up to t insertions or deletions. Furthermore, he showed that the Varshamov-Tenengolts codes [VT65] can be used to correct a single insertion or deletion error. Notably, most of the classic techniques for constructing error-correcting codes correcting substitutions or erasures cannot be applied to construct codes coping with insertions and deletions. This is due to the loss in synchronization between sender and receiver caused by these types of errors. For an overview on recent developments of new approaches and methods for the deletion channel we refer to [CR20].

We first recap known results for adversarial deletion channels without the presence of noiseless feedback to the sender. Let a code $\mathcal{C} \subseteq \{0,1\}^n$ consist of 2^{Rn} binary

codewords of length n. The design goal is to construct a code of maximal rate that is capable of correcting a fraction of τ *adversarial* deletions. The term adversarial deletions in this context means that an adversary knowing encoding and decoding strategies as well as the message being sent can inflict an arbitrary pattern consisting of t deletions. Consequently, $n - t$ symbols pass the channel unaltered.

The error correction capability of $\mathcal{C}$ is strongly related to the notion of the longest common subsequence between two codewords. For a code being able to correct a fraction of τ errors the longest common subsequence between any two codewords in $\mathcal{C}$ is of length less than $(1 - \tau)n$. It is clear that for $\tau \geq \frac{1}{2}$, the adversary can force the channel to output either the all-one word or the all-zero word of length $n/2$. Thus, a code $\mathcal{C}$ can contain at most two codewords and its rate is zero. The first code construction capable of correcting a non-zero fraction of deletions was proposed by Schulman and Zuckerman [SZ99]. They proposed to use concatenated codes composed of non-binary outer codes and well-performing short binary inner codes that can be found by brute-force search. The most recent improvement in this direction is due to Bukh, Guruswami, and Håstad [BG16; BGH16]. Their work provides a family of code constructions with a positive rate for any $\tau < \sqrt{2} - 1 \approx 0.41$. For the adversarial substitution channel it is well known [Plo60] that it is not possible to have exponentially many codewords in a code tolerating a fraction of $\tau \geq 0.25$ errors. However, to the best of our knowledge not much further progress has been made on the limitations of deletion-correcting codes. Even determining the maximal fraction of deletions for which there exist codes with rate bounded away from zero remains an open research problem.

In this chapter we determine the capacity error function of the insertion-deletion channel with noiseless instantaneous feedback. Our results show that as for the synchronized channels examined in Chapter 6 and Chapter 7, noiseless feedback can enhance the capacity error function for the insertion-deletion channel.

This problem is relevant in practice because certain channels, e.g. DNA-based storage channel, are prone to inflict insertion-deletion errors rather than substitution errors. Throughout this chapter we only consider the binary alphabet at the channel input. The generalization to non-binary alphabets is not straightforward and remains an interesting open research question.

For any message $m \in \mathcal{M}$, the sender's task is to encode m into a binary string $\boldsymbol{c} \in \{0, 1\}^n$ such that after transmitting this string over the insertion-deletion channel the message m can be recovered without error. More specifically m needs to be successfully reconstructed for any error pattern composed of at most $\lceil \tau n \rceil$ insertion or deletion errors. Recalling Remark 5.1 this coding strategy is capable of dealing with an adversary knowing m and the coding strategy that inflicts up to $\lceil \tau n \rceil$ insertion or deletion errors (mixing of insertions and deletions is permitted as long as their total number is upper bounded by $\lceil \tau n \rceil$). The following explanations take the view of transmitting data over an adversarial channel.

We denote the channel's output string after the transmission of the entire block by $\boldsymbol{y} \in \{0,1\}^*$ and denote the channel output corresponding to the i-th codeword symbol c_i by $\boldsymbol{y}_i \in \{0,1\}^*$. This output can be created arbitrarily by the adversary using insertions and/or deletions in accordance to the remaining error budget for the respective block transmission. We stress that the adversary has the ability to place the insertions before the symbol c_i and not only after the transmitted symbol c_i, increasing his flexibility. The decoder only gets access to $\boldsymbol{y} = (\boldsymbol{y}_1 || \dots || \boldsymbol{y}_n)$ but not its partitioning, i.e. the decoder is not able to split the output string into the respective $\boldsymbol{y}_i$. This means that the decoder does not know which symbols within the received block $\boldsymbol{y}$ belong to which encoded symbol c_i. This is in contrast to the Z-channel and the unidirectional channels for which input and output symbols are synchronized.

However, due to the feedback the encoding algorithm has access to the tuple $\hat{\boldsymbol{y}}^{(i)} := (\boldsymbol{y}_1, \dots, \boldsymbol{y}_i) \in (\{0,1\}^*)^i$ and hence is able to split the output sequence into its components before deciding on the $(i+1)$st encoded symbol c_{i+1}. In particular it holds that $c_{i+1} : \mathcal{M} \times (\{0,1\}^*)^i \to \{0,1\}$ is a function of m and $\hat{\boldsymbol{y}}^{(i)}$, i.e., $c_{i+1} = c_{i+1}(m, \hat{\boldsymbol{y}}^{(i)})$.

The number of errors that the adversary can inflict during the block transmission is upper bounded by $t = \lceil \tau n \rceil$. Formalizing this statement it holds that

$$\sum_{i=1}^{n} \Delta(c_i(m, \hat{\boldsymbol{y}}^{(i-1)}), \boldsymbol{y}_i) \leq t \ . \tag{8.1}$$

We denote the insertion-deletion channel with binary input by Γ_{id}^2 and maximal cardinality of the message set for which a successful coding strategy exists by $M_2(\Gamma_{id}^2, n, t)$. In accordance with Definition 5.3 the capacity error function for the insertion-deletion channel is specified by

$$C_2^f(\Gamma_{id}^2, \tau) := \limsup_{n \to \infty} \frac{\log_2(M_2(\Gamma_{id}^2, n, \lceil \tau n \rceil))}{n} \ . \tag{8.2}$$

Notice that the length of the channel output for the insertion-deletion channel with parameter τ varies between $n - \lceil \tau n \rceil$ and $n + \lceil \tau n \rceil$ for a blocklength of n input symbols.

8.1.1 Main Results

In this chapter, we show that the problem of transmitting information over the adversarial insertion-deletion channel can be reduced to the problem of transmitting information over the adversarial substitution channel. More specifically, we first demonstrate that any encoding algorithm with n transmissions tackling t insertions can serve as an encoding algorithm with $n + t$ transmissions correcting t substitution errors. From that result we obtain an upper bound on $C_2^f(\Gamma_{id}^2, \tau)$. Then we adapt an encoding strategy originally suggested for the feedback substitution channel to obtain a lower bound on $C_2^f(\Gamma_{id}^2, \tau)$. This lower bound matches the previously derived

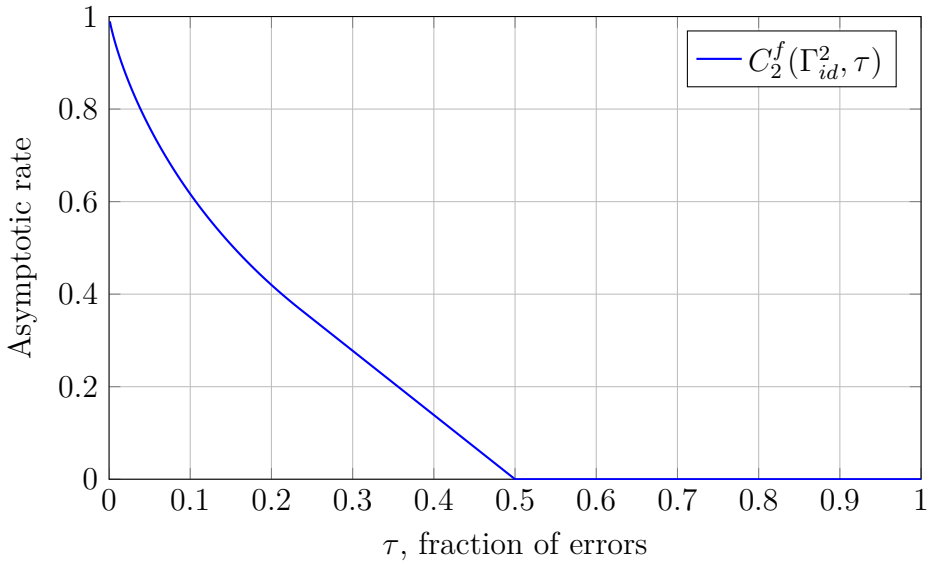

Figure 8.1.1: Capacity error function of the insertion-deletion channel with noiseless feedback.

upper bound. Therefore, we have determined $C_2^f(\Gamma_{id}^2, \tau)$ for any fraction of errors τ, $0 \leq \tau \leq 1$.

Theorem 8.1. *The capacity error function for the adversarial insertion-deletion channel utilizing noiseless feedback is*

$$C_2^f(\Gamma_{id}^2, \tau) = \begin{cases} (1+\tau)\left(1 - h\left(\frac{\tau}{1+\tau}\right)\right), & \text{for } 0 \leq \tau \leq \sqrt{5}-2, \\ (1-2\tau)\log\left(\frac{1+\sqrt{5}}{2}\right), & \text{for } \sqrt{5}-2 < \tau \leq \frac{1}{2}, \\ 0, & \text{otherwise.} \end{cases} \tag{8.3}$$

The resulting asymptotic rate is plotted in Figure 8.1.1. As a side contribution in this chapter, we present a more elaborate version of Zigangirov's technical analysis [Zig76] of Horstein's algorithm [Hor63] for the adversarial substitution channel. We hope that this makes the intuitive algorithm, which is nevertheless hard to analyze, more accessible to a wider audience.

8.1.2 Outline

The remainder of the chapter is organized as follows. In Section 8.2, we provide an upper bound on $C_2^f(\Gamma_{id}^2, \tau)$. In Section 8.3, we discuss a feedback insertion-deletion code and its reduction to feedback substitution codes. We conclude this chapter with Section 8.4. The analysis of the feedback substitution code is given in Appendix 8.A.

8.2 Upper Bound on $C_2^f(\Gamma_{id}^2, \tau)$

In this section we first recap the known capacity error function of the symmetric channel. Then we prove that a feedback insertion-deletion code correcting a fraction of τ errors can serve as a feedback substitution code capable of correcting a fraction of $\tau/(1+\tau)$ errors and use this result to prove an upper bound on $C_2^f(\Gamma_{id}^2, \tau)$.

8.2.1 Construction of a Substitution Code from an Insertion-Deletion Code

To prove that (8.3) is an upper bound on the maximal rate of feedback codes for the adversarial insertion-deletion channel, we combine two ideas. First, we prove that a feedback insertion-deletion code can be used to tackle substitution errors.

Lemma 8.1. *Suppose that a feedback insertion-deletion code of length n and size M capable of correcting t errors is given. Then there exists a feedback code of length $n+t$ and size M capable of correcting t substitution errors.*

Second, we make use of $C_2^f(\Gamma_S^2, \tau)$ established by Berlekamp [Ber64] and Zigangirov [Zig76] and given in Theorem 5.1. By Lemma 8.1, we have $M_S(n+t, t) \geq M_{id}(n, t)$ and, thus, $(1+\tau)C_2^f(\Gamma_S^2, \tau/(1+\tau)) \geq C_2^f(\Gamma_{id}^2, \tau)$. This bound and Theorem 5.1 yield that the asymptotic rate specified in Theorem 8.1 is indeed an upper bound on the maximal achievable rate of the adversarial insertion-deletion channel. It remains to prove Lemma 8.1 which we do in the following.

Proof of Lemma 8.1. Let the encoding functions of the given feedback insertion-deletion code be denoted by $c_i : \mathcal{M} \times (\{0,1\}^*)^{i-1} \to \{0,1\}$ for $i \in [n]$ and let the decoding function be given as $dec : \{0,1\}^* \to \mathcal{M}$. We define the encoding functions $e_j : \mathcal{M} \times \{0,1\}^{j-1} \to \{0,1\}$ of the symmetric channel Γ_S^2 for $j \in [n+t]$ using the encoding functions c_i as components. To prove the statement our goal is to show that the feedback coding strategy specified by the e_j is successful for Γ_S^2 for any pattern of at most t errors.

We denote the number of correctly received symbols by i, the number of transmitted symbols by j, the index of the last correctly received symbol by k, the received output string by $\boldsymbol{z}$, and the tuple of output strings according to the output of a potential insertion-deletion channel by $\hat{\boldsymbol{y}}$.

The encoding scheme we propose consists of three parts. The Initialisation step is for initializing counters and state variables. Step 1 describes the process of re-transmitting symbols c_i until they have been received correctly. Step 2 is just used to fill potentially unused symbols within the block with zero symbols. The state variables i and k are only updated in Step 1.

Initialisation: Let $i \leftarrow 0$, $j \leftarrow 0$, $k \leftarrow 0$, $\boldsymbol{z} \leftarrow (\,)$, and $\hat{\boldsymbol{y}} \leftarrow (\,)$.

Step 1: Define and transmit $e_{j+1}(m, \boldsymbol{z}) = c_{i+1}(m, \hat{\boldsymbol{y}})$. Update $j \leftarrow j + 1$. Let z_j be the received symbol. Update $\boldsymbol{z} \leftarrow (\boldsymbol{z}||z_j)$. If $z_j \neq e_j(m, \boldsymbol{z})$, then repeat Step 1. Otherwise, update $i \leftarrow i + 1$, define $\boldsymbol{y}_i \leftarrow \boldsymbol{z}_{[k+1,j]}$, update $\hat{\boldsymbol{y}} \leftarrow (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_i)$ and $k \leftarrow j$. If $i = n$, then go to Step 2. Otherwise, repeat Step 1.

Step 2: If $j = n + t$, update $\boldsymbol{y}_n \leftarrow (\boldsymbol{y}_n||\boldsymbol{z}_{[k+1,n+t]})$, $\hat{\boldsymbol{y}} \leftarrow (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ and exit algorithm. Otherwise, define and transmit $e_{j+1}(m, \boldsymbol{z}) = 0$. Update $j \leftarrow j + 1$. Let z_j be the received symbol. Update $\boldsymbol{z} \leftarrow (\boldsymbol{z}||z_j)$. Repeat Step 2.

The encoding algorithm successfully terminates after $n+t$ transmissions because we will have n correct receptions within a block of $n + t$ symbols. The decoding scheme can just be taken from the insertion-deletion code because the output string $\boldsymbol{z}$ is a possible output of the insertion-deletion channel with at most t errors for encoding functions c_i, $i \in [n]$. Indeed, if we define $\hat{\boldsymbol{y}}^{(i)} := (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_i)$, we obtain

$$\sum_{i=1}^{n} \Delta(c_i(m, \hat{\boldsymbol{y}}^{(i)}), \boldsymbol{y}_i) = \sum_{i=1}^{n} (|\boldsymbol{y}_i| - 1) = \sum_{i=1}^{n} |\boldsymbol{y}_i| - n = (n + t) - n = t.$$

Therefore, the decoder of the insertion-deletion code outputs the correct message, i.e., $dec(\boldsymbol{z}) = m$. This completes the proof. ■

8.3 Lower Bound on $C_2^f(\Gamma_{id}^2, \tau)$

In this section, we provide a feedback insertion-deletion code with an asymptotic rate arbitrarily close to (8.3). We adapt an algorithm originally suggested by Horstein [Hor63] and further developed by Schalkwijk [Sch71] and Zigangirov [Zig76]. Note that this algorithm has already been used for the adversarial substitution channel with feedback. We point out that not every encoding strategy for the adversarial substitution channel can be adapted to be used for the insertion-deletion channel. A key property of Horstein's algorithm is that in case the channel inflicts an error the sender retransmits the intended symbol until it is received correctly. We elaborate on the importance of this property in the following. Consider an arbitrary encoding strategy for the insertion-deletion channel with noiseless feedback. We assume that according to the previously received symbols the encoding strategy implies sending $c_i = 0$ over the channel and that the adversary inserts a "1" symbol before c_i such that $\boldsymbol{y}_i = 10$. Imagine that this "1" symbol was inserted at the end of $\boldsymbol{y}_{i-1}$ instead. If observing $\boldsymbol{y}_{i-1}$ in this case lead the encoder to change the i-th symbol to $c_i = 1$, the adversary would be able to inflict multiple errors at the price of inflicting a single insertion. For encoding schemes that retransmit the same symbol again after observing incorrect reception via the feedback, e.g., Horstein's encoding procedure, this is not the case.

8.3.1 Feedback Insertion-Deletion Code

Suppose that at most a fraction of τ errors can occur and the sender would like to transmit a message $m \in \mathcal{M}$, where $M := |\mathcal{M}| = 2^{Rn}$, where $R = C_2^f(\Gamma_{id}^2, \tau) - \varepsilon$ and $\varepsilon > 0$. We enumerate $\mathcal{M}$ and identify its elements by $[M]$. Based on τ, ε and n, the sender and the receiver choose two non-algebraic numbers α and β fulfilling $\alpha + \beta = 2$. For $\tau < \sqrt{5} - 2$, they take α sufficiently close to $2\tau/(1+\tau)$ and for $\sqrt{5} - 2 \leq \tau < \frac{1}{2}$, α is chosen to be around $(3 - \sqrt{5})/2$. We discuss how close this parameter has to be taken in Section 8.A.

Remark 8.1. *We remark that the non-algebraic numbers are dense within the set of real numbers, meaning that we can approximate any real number arbitrarily close within the non-algebraic numbers.*

The sender divides the $[0, 1]$-segment into M subsegments of length $1/M$ and enumerates them from left to right by the elements of $[M]$. The length of these M segments will vary during transmission. If the sender wishes to transmit the message $m \in [M]$, then the m-th segment will be called the *true segment*. Let $T(i)$ denote the true segment of length $t(i)$ after the i-th channel use. Denote the union of segments left and right of $T(i)$ by $L(i)$ and $R(i)$, respectively. We label the coding scheme for the insertion-deletion channel by $\mathfrak{A}_{\mathfrak{id}}(M, n, \alpha)$ and describe its encoding and decoding procedures in the following.

Encoding Procedure: Before sending the i-th symbol c_i over the channel, the sender checks whether the center of the true segment lies in $[0, \frac{1}{2})$. If this is the case, the encoder transmits the symbol $c_i = 0$ over the channel, otherwise the encoder transmits $c_i = 1$. The sender observes the channel outputs $\boldsymbol{y}_i$ of length n_i and modifies the length of all segments in the following manner. The sender runs over the symbols of $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,n_i}) \in \{0,1\}^{n_i}$ from left to right. If $\boldsymbol{y}_i$ is the empty string, then the next symbol is transmitted. Otherwise, if $y_{i,j} = 0$, then the length of all segments, that entirely lie inside $[0, \frac{1}{2})$, is enlarged by a factor of β. The length of segments that lie entirely in $[\frac{1}{2}, 1]$ is multiplied (reduced) by the factor α. Usually, there is a segment that contains the point $\frac{1}{2}$ (not as its endpoint). Let x be the length of this segment. Then x can be represented as $x = x_0 + x_1$, where x_0 is the length of part of the segment, which lies in $[0, \frac{1}{2})$. Then the length of this segment is updated from x to $\beta x_0 + \alpha x_1$. If $y_{i,j} = 1$ is received, the encoder shrinks the segments left to the point $\frac{1}{2}$ by a factor of α while increasing the segments right to it by a factor of β. Basically, compared to the case $y_{i,j} = 0$ the roles of α and β are just swapped. During the transmissions the lengths of the segments change according to the algorithm, however, the order of the segments remains unaltered. In total, the sender makes n_i such updates after receiving the output sequence $\boldsymbol{y}_i$ corresponding to the i-th channel use. It can be readily seen that the total length of all segments is always 1. The algorithm is designed to increase the length of the true segment such that it contains the point $\frac{1}{2}$ after the last transmission.

The complexity of this encoding algorithm is $\Theta(Mn)$. However, it is clear from the description the encoder only needs to keep track of $L(i)$, $T(i)$ and $R(i)$. Thus, the complexity can be reduced to $\Theta(n)$.

Decoding Procedure: After receiving a string $\boldsymbol{y} = (y_1||\dots||y_{n'})$ of length $n' \in [n-t, n+t]$, the decoder is able to reconstruct the sender's point of view by varying the length of all segments in the same way as the sender. After n' steps, the decoder outputs the message corresponding to the segment containing the point $\frac{1}{2}$. The complexity of this decoding algorithm is $\Theta(Mn)$. To reduce the decoding complexity the receiver can reverse the decoding procedure in the following way. The decoder runs over the symbols of $\boldsymbol{y}$ from right to left and tracks the preimage of the point $\frac{1}{2}$. Formally, let $P(n') := \frac{1}{2}$. For $j \in \{0, 1, \dots, n'-1\}$ and $y_j = 0$, define

$$P(j-1) := \begin{cases} \frac{P(j)}{\beta}, & \text{if } P(j) \leq \frac{\beta}{2}, \\ \frac{1}{2} + \frac{2P(j)-\beta}{2\alpha}, & \text{if } P(j) > \frac{\beta}{2}. \end{cases}$$

For $y_j = 1$, define

$$P(j-1) := \begin{cases} \frac{P(j)}{\alpha}, & \text{if } P(j) \leq \frac{\alpha}{2}, \\ \frac{1}{2} + \frac{2P(j)-\alpha}{2\beta}, & \text{if } P(j) > \frac{\alpha}{2}. \end{cases}$$

Finally, the decoder uses $\hat{m} := \lceil M \cdot P(0) \rceil$ as an estimate for the message m that the encoder intended to send. Clearly, the complexity of this simplified algorithm is also $\Theta(n)$.

8.3.2 Intuition Behind the Algorithm

We give an intuitive explanation why this strategy is sensible. In this section we are only considering the case that $\tau < \sqrt{5} - 2$ as only for this case

$$C_2^f(\Gamma_{id}^2, \tau) = (1+\tau)\left(1 - h\left(\frac{\tau}{1+\tau}\right)\right).$$

Let us take $\alpha = 2\tau/(1+\tau)$. Assume that during the encoding procedure the number of times when the true segment contains the point $\frac{1}{2}$ is $o(n)$ and the number of insertion and deletion errors during transmission is $\tau_{ins}n + o(n)$ and $\tau_{del}n + o(n)$, respectively, where $\tau_{ins} + \tau_{del} \leq \tau$. Then the length of the true segment after n transmissions, $t(n)$, can be bounded as follows

$$\begin{aligned} t(n) &\geq M^{-1}\alpha^{\tau_{ins}n+o(n)}\beta^{(1-\tau_{del})n+o(n)} \\ &= 2^{-(C_2^f(\Gamma_{id}^2,\tau)-\varepsilon)n+\tau_{ins}n\log(2\tau/(1+\tau))+(1-\tau_{del})n\log(2/(1+\tau))+o(n)} \\ &\geq 2^{-(C_2^f(\Gamma_{id}^2,\tau)-\varepsilon)n+\tau n\log(2\tau/(1+\tau))+n\log(2/(1+\tau))+o(n)} \\ &= 2^{\varepsilon n+o(n)}. \end{aligned}$$

where we used the fact that the function $\tau_{ins}\log(2\tau/(1+\tau)) + (1-\tau_{del})\log(2/(1+\tau))$ provided $\tau_{ins} + \tau_{del} \leq \tau$ is minimized at $\tau_{ins} = \tau$ and $\tau_{del} = 0$. Thus, the length of the true segment would be very large at the end and it would definitely contain the point $\frac{1}{2}$. However, our initial assumption is wrong, and the true segment contains the point $\frac{1}{2}$ more than $o(n)$ times. To actually prove that the strategy is working, we need more sophisticated arguments.

8.3.3 Construction of an Insertion-Deletion Code from a Substitution Code

The algorithm $\mathfrak{A}_{\mathfrak{id}}(M, n, \alpha)$ described in Section 8.3.1 was originally proposed in a similar form for the substitution channel. We emphasize that this channel provides synchronization between the sender and the receiver, i.e., after transmitting the symbol c_i, the channel outputs exactly one symbol y_i. We will refer to the encoding algorithm for the substitution channel with n channel uses that follows the steps described in Section 8.3.1 as $\mathfrak{A}_s(M, n, \alpha)$. A formal definition of this algorithm for the substitution channel will be given in Section 8.A.1. In the following we show that it is possible to achieve the asymptotic rate given in equation (8.3) for the adversarial insertion-deletion channel by using the algorithm $\mathfrak{A}_{\mathfrak{id}}(M, n, \alpha)$. We first prove the following Lemma.

Lemma 8.2. *If for all $n' \in [n-t, n+t]$, the algorithm $\mathfrak{A}_s(M, n', \alpha)$ can be used for successful transmissions of any message $m \in \mathcal{M}$ over the feedback adversarial substitution channel with at most $t' := \lfloor (n'-n+t)/2 \rfloor$ errors, then the algorithm $\mathfrak{A}_{id}(M, n, \alpha)$ can be used for successful transmissions of any message $m \in [M]$ over the feedback adversarial insertion-deletion channel with at most t errors.*

Proof. Let $\boldsymbol{y} = (\boldsymbol{y}_1 || \ldots || \boldsymbol{y}_n) = (y_1, \ldots, y_{n'})$ be a possible output string of length $n' \in [n-t, n+t]$ if the algorithm $\mathfrak{A}_{id}(M, n, \alpha)$ is used for transmitting a message $m \in \mathcal{M}$ over the adversarial insertion-deletion channel with at most t errors. Now we show that this string $\boldsymbol{y}$ is also a possible output if the algorithm $\mathfrak{A}_s(M, n', \alpha)$ is used for transmitting the same message $m \in \mathcal{M}$ over the adversarial substitution channel with at most $t' := \lfloor (n'-n+t)/2 \rfloor$ errors. Define $\hat{\boldsymbol{y}}^{(i-1)} := (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{i-1}) \in (\{0,1\}^*)^{i-1}$ and $\boldsymbol{y}^{(i-1)} := (y_1, \ldots, y_{i-1}) \in \{0,1\}^{i-1}$. Let $c_i = c_i(m, \hat{\boldsymbol{y}}^{(i-1)})$ and $e_i = e_i(m, \boldsymbol{y}^{(i-1)})$ be the ith bit generated by the encoders of the feedback insertion-deletion code and the substitution code, respectively. We define $\boldsymbol{e} := (e_1, \ldots, e_{n'})$. So, it suffices to show that $d_H(\boldsymbol{e}, \boldsymbol{y}) \leq t'$. Denote the length of $\boldsymbol{y}_i$ by n_i. We represent $\boldsymbol{e}$ as the concatenation $(\boldsymbol{e}_1 || \ldots || \boldsymbol{e}_n)$ such that the binary string $\boldsymbol{e}_i$ has length n_i. It is clear that for "synchronized" moments i and $N_i := n_1 + \cdots + n_{i-1} + 1$, both algorithms send the same symbol, i.e.,

$$c_i(m, \hat{\boldsymbol{y}}^{(i-1)}) = e_{N_i}(m, \boldsymbol{y}^{(N_i-1)}).$$

By the definition of the algorithm, it can be readily checked that if $y_j \neq e_j$, then $e_{j+1} = e_j$, i.e., a re-transmission occurs. Therefore,

$$d_H(\boldsymbol{e}_i, \boldsymbol{y}_i) \leq \begin{cases} n_i - 1, & \text{if } \Delta(c_i, \boldsymbol{y}_i) = n_i - 1, \\ n_i, & \text{if } \Delta(c_i, \boldsymbol{y}_i) = n_i + 1. \end{cases}$$

Note that the second case can only happen when a deletion occurred. Let t_{ins} be the total number of insertions and t_{del} be the total number of deletions. We have the restrictions $t_{ins} + t_{del} \leq t$ and $n - t_{del} + t_{ins} = n'$. This implies that $n' - n + 2t_{del} \leq t$ or $t_{del} \leq \lfloor (n - n' + t)/2 \rfloor$. It follows that

$$d_H(\boldsymbol{e}, \boldsymbol{y}) = \sum_{i=1}^{n} d_H(\boldsymbol{e}_i, \boldsymbol{y}_i) \leq \sum_{i=1}^{n} n_i - \sum_{i=1}^{n} \mathbb{1}\{\Delta(c_i, \boldsymbol{y}_i) = n_i - 1\} \leq n' - (n - t_{del}) \leq t',$$

where $\mathbb{1}\{x = a\}$ denotes the indicator function of the event $x = a$. ■

Next we have a closer look at the adversarial substitution channel. In order to achieve the capacity error function of the adversarial substitution channel with a fraction of $\tau \leq (3 - \sqrt{5})/4$ errors, for the parameter α within the algorithm $\mathfrak{A}_s(M, n, \alpha)$ it holds that $\alpha \approx 2\tau$ with α being non-algebraic.

Lemma 8.3 (Follows from [Zig76])**.** *Let $\alpha^* = 2\tau^*$ with $\tau^* \leq (3 - \sqrt{5})/4$ being a non-algebraic number and let the algorithm $\mathfrak{A}_s(M, n, \alpha^*)$ be used to communicate information over the adversarial substitution channel for different τ. Then the achieved rates for different τ form a tangent line to the point $(\tau^*, C_2^f(\Gamma_s^2, \tau^*))$ of the function $C_2^f(\Gamma_s^2, \tau)$.*

By using Lemma 8.3 we can show the following monotonicity property.

Proposition 8.1. *Let $M_s(n, \tau, \alpha)$ denote the maximum number of messages M that can successfully be transmitted using the algorithm $\mathfrak{A}_s(M, n, \alpha)$ for the adversarial substitution channel with blocklength n and at most τn errors. Let α and τ be non-algebraic such that $\alpha = 2\tau/(1 + \tau)$ and $\alpha \leq (3 - \sqrt{5})/2$. Then it holds that*

$$\begin{aligned} \min_{k \in \{0,1,\ldots,\tau n\}} \log M_s(\lfloor n(1+\tau) \rfloor - 2k, \lfloor \tau n \rfloor - k, \alpha) &= (1 + o(1)) \log M_s(\lfloor (1+\tau)n \rfloor, \lfloor \tau n \rfloor, \alpha) \\ &= n(1+\tau) C_2^f \left(\Gamma_s^2, \left(\frac{\tau}{1+\tau} \right) \right) + o(n). \end{aligned}$$

Proof. We fix $\alpha = 2\tau/(1 + \tau)$ because it allows us to successfully transmit by the algorithm $\mathfrak{A}_s(M, \lfloor n(1+\tau) \rfloor, \alpha)$ asymptotically the maximum amount of messages given the number of errors is at most $\lfloor \tau n \rfloor$. Then our goal is to show that although the chosen α is not optimal in terms of the achievable rate for blocklength $\lfloor n(1 + \tau) \rfloor - 2k$ and at most $\lfloor \tau n \rfloor - k$ errors, the quantity $\log M_s(\lfloor n(1 + \tau) \rfloor - 2k, \lfloor \tau n \rfloor - k, \alpha)$ is still asymptotically minimized for $k = 0$.

According to Lemma 8.3 by fixing $\alpha = 2\tau/(1+\tau)$ and using the algorithm $\mathfrak{A}_s(M, n', \alpha)$ for $n' = \lfloor n(1+\tau) \rfloor - 2k$ and at most $\lfloor \tau n \rfloor - k$ errors the achievable rates form the tangent line to the point $(\tau/(1+\tau), R_s(, \tau/(1+\tau)))$ of the function $R_s(x) := 1 - h(x)$.

To prove the required claim concerning the minimization, we define the function

$$f(x) := \frac{\log M_s(n(1+\tau-2x), (\tau-x)n, \alpha)}{n} = (1+\tau-2x)\left(a\frac{\tau-x}{1+\tau-2x} + b\right)(1+o(1)),$$

where the real valued a, b are fixed by the requirement that the achieved rates form a tangent to the point $(\tau/(1+\tau), R_s(\tau/(1+\tau)))$. A short computation shows that $a = \log \tau$ and $b = 1 - \log(1+\tau)$.

The derivative of $f(x)$ as $n \to \infty$ is

$$\frac{\partial f}{\partial x} = -a - 2b + o(1) = \log\left(\frac{(1+\tau)^2}{\tau}\right) - 2 + o(1) > 0,$$

where the inequality follows the argument of the logarithm is strictly greater than 4 for $\tau < 1/2$. This shows that f is asymptotically minimized for $x = 0$, completing the proof. ■

We have already shown that the output sequences of the algorithm $\mathfrak{A}_{id}(M, n, \alpha)$ for the adversarial insertion-deletion channel are also within the set of output sequences of the adversarial substitution channel if the algorithm $\mathfrak{A}_s(M, n', \alpha)$ is used for at most $t' = \lfloor (n' - n + t)/2 \rfloor$ errors and $n' \in [n-t, n+t]$. Therefore, from Proposition 8.1 it follows that $R_{id}(\tau) \geq (1+\tau)R_s(\tau/(1+\tau))$. By Theorem 5.1 it holds that the rate $R_{id}(\tau)$, given in (8.3), is actually achievable and upper and lower bounds on the adversarial insertion-deletion channel coincide.

8.4 Summary and Open Problems

In this chapter, we have introduced a new problem statement of transmitting information through the adversarial insertion-deletion channel with feedback. We have shown a reduction of this problem to a problem of communication over the adversarial substitution channel. Thereby, capacity error function for the adversarial insertion-deletion channel with noiseless feedback has been established. In particular, the rate is positive whenever the fraction of insertion and deletion errors inflicted by the channel is less than $\frac{1}{2}$. We also revisit Horstein's algorithm [Hor63] for the adversarial substitution channel with feedback and present a more elaborate version of Zigangirov's analysis [Zig76].

We emphasize that all results discussed in the chapter concern the binary channel. A natural question that arises is whether it is possible to extend the methodology for the q-ary case. The best currently known results for the adversarial q-ary substitution

channel with feedback are discussed in [Leb16]. In particular, for $0 < \tau < 1/q$, the capacity error function is only established for a countable number of values for τ. Moreover, we point out that it is hard to generalize Zigangirov's arguments to the q-ary case and, in particular, find appropriate analogues of functions u and v defined in Appendix 8.A.

Appendix

8.A Analysis of the Feedback Substitution Code

In this section, we revisit the algorithm and the proof suggested by Zigangirov in [Zig76]. To make this section self-contained, we first describe an encoding process for the adversarial substitution channel with feedback. Then we introduce useful concepts and give a high-level analysis of this feedback strategy. Finally, we provide proofs of auxiliary technical statements used for showing the main result.

8.A.1 Feedback Substitution Code

Suppose that at most τn substitution errors can occur during n transmissions and the sender wants to transmit one out of M messages. If $0 \leq \tau < (3-\sqrt{5})/4$, then the encoder takes positive real values α and β such that $\alpha + \beta = 2$ and α is sufficiently close to 2τ. If $(3-\sqrt{5})/4 \leq \tau \leq 1/3$, then α and β are selected such that $\alpha + \beta = 2$ and α is sufficiently close to but less than $(3-\sqrt{5})/2$. Clearly, for this assignment, we have that

$$\alpha\beta^2 \leq 1\,. \tag{8.4}$$

Encoding Procedure: Initially, the sender takes the $[0,1]$-segment and partitions it into M subsegments of equal length. Subsegments are enumerated from left to right. The jth message, $j \in [M]$, is associated with the jth segment. Suppose that the encoder wants to transmit a message $m \in [M]$. The mth segment is then called the *true* segment and we denote it by $T(i)$ after the ith transmission. We define the union of the segments on the left and on the right to $T(i)$ by $L(i)$ and $R(i)$, respectively. The lengths of $L(i)$, $T(i)$ and $R(i)$ are denoted by $l(i)$, $t(i)$ and $r(i)$. Clearly, $t(0) = 1/M$ and $l(i) + t(i) + r(i) = 1$. At the ith moment, the sender checks whether the center of $T(i-1)$ is less than $\frac{1}{2}$ and transmits a "0" to the channel in that case. Otherwise the encoder transmits a "1" to the channel. If a "0" is received, then everything on the left to $\frac{1}{2}$ is enlarged by β and everything on the right to $\frac{1}{2}$ is shrunk by α. More

formally,

$$\begin{aligned} l(i) &:= \beta \min\left(\frac{1}{2}, l(i-1)\right) + \alpha \max\left(l(i-1) - \frac{1}{2}, 0\right), \\ r(i) &:= \alpha \min\left(\frac{1}{2}, r(i-1)\right) + \beta \max\left(r(i-1) - \frac{1}{2}, 0\right), \\ t(i) &:= 1 - l(i) - r(i). \end{aligned}$$

If a "1" is received, then everything on the left to $\frac{1}{2}$ is shrunk by α and everything on the right to $\frac{1}{2}$ is enlarged by β. Thus,

$$\begin{aligned} l(i) &:= \beta \max\left(l(i-1) - \frac{1}{2}, 0\right) + \alpha \min\left(\frac{1}{2}, l(i-1)\right), \\ r(i) &:= \alpha \max\left(r(i-1) - \frac{1}{2}, 0\right) + \beta \min\left(\frac{1}{2}, r(i-1)\right), \\ t(i) &:= 1 - l(i) - r(i). \end{aligned}$$

Informally, given $\varepsilon > 0$, for properly chosen $\alpha \approx 2\min(\tau, (3-\sqrt{5})/4)$, sufficiently large n and $M = 2^{(R_s(\tau)-\varepsilon)n}$, we claim that $T(n)$ contains the point $\frac{1}{2}$.

8.A.2 Preliminaries

Define $x(i) := \min(l(i), r(i))$ and $y(i) := \max(l(i), r(i))$. We say $T(i)$ is *central* if $y(i) \leq \frac{1}{2}$ and we say that there is a *crossing* between moment $i-1$ and moment i if either (1) $l(i-1) \leq r(i-1)$ and $l(i) > r(i)$ or (2) $l(i-1) > r(i-1)$ and $l(i) \leq r(i)$. We say $T(i)$ is *balanced* if $y(i)/x(i) \leq \beta/\alpha$. We refer to Figure 8.A.1 for an illustration of the properties of $T(i)$.

Example 8.1. *In Figure 8.A.2, one can see how the segments $L(i), R(i)$, and $T(i)$ with $l(i) = 0.2$, $r(i) = 0.4$ and $t(i) = 0.4$ change for the case of a correct and incorrect transmission when using the parameters $\tau = 0.15$, $\alpha = 2\tau$, and $\beta = 2 - \alpha$. In particular, we see that the true segment $T(i)$ is the one containing the point $\frac{1}{2}$ and, thus, is central. Moreover, we see that $x(i) = l(i)$ and $y(i) = r(i)$, and can conclude that $T(i)$ is also balanced as $2 = \frac{y(i)}{x(i)} \leq \frac{\beta}{\alpha} = \frac{1.7}{0.3}$. In the example, a correct transmission is sending a "0", i.e., enlarging the intervals on the left of $\frac{1}{2}$ by β and shrinking the right hand side by α. We can see that there was a crossing, since $l(i) \leq r(i)$ and now $l(i+1) > r(i+1)$. Moreover, the true segment at moment $i+1$ still covers the point $\frac{1}{2}$ and $T(i+1)$ is still balanced. In contrast, if the transmission was incorrect, i.e., enlarging by β the right hand side of $\frac{1}{2}$ and shrinking the other part by α, we observe that there was no crossing and the true segment $T(i+1)$ does not contain the point $\frac{1}{2}$ anymore. Moreover, we have that $T(i+1)$ is not balanced, since $\frac{34}{3} = \frac{y(i)}{x(i)} > \frac{\beta}{\alpha} = \frac{1.7}{0.3}$.*

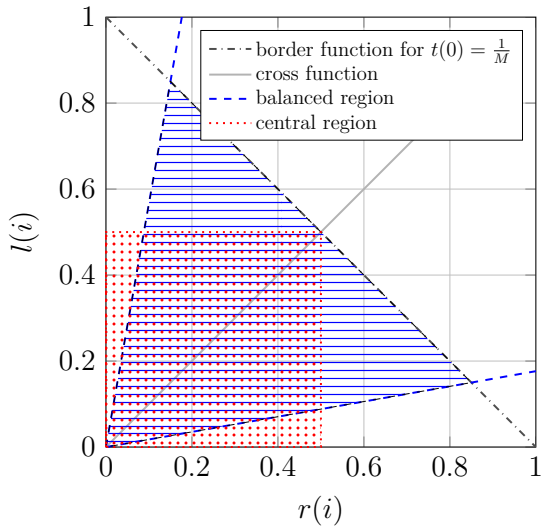

Figure 8.A.1: Illustration for the notions of a crossing, a balanced and central true segment and the corresponding areas when $\tau = 0.15, \alpha = 2\tau$, and $\beta = 2 - \alpha$.

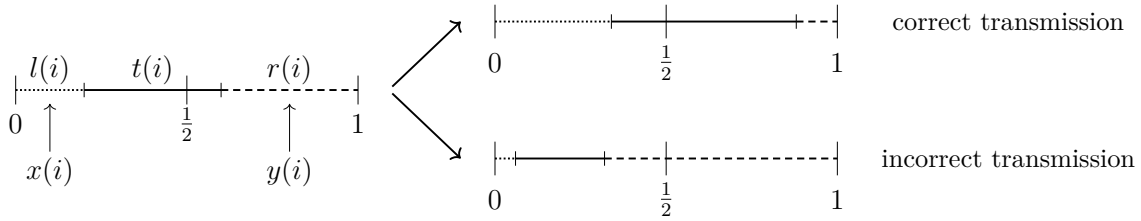

Figure 8.A.2: Example for transition of the intervals $L(i), T(i)$, and $R(i)$ in case for a correct and incorrect transmission

Define the function

$$g(i_0, i_1) := e(i_0, i_1) \log \alpha + f(i_0, i_1) \log \beta \,,$$

where $e(i_0, i_1)$ and $f(i_0, i_1)$ are the numbers of incorrect and correct transmissions in moments $(i_0, i_1] = \{i_0 + 1, \ldots, i_1\}$. Note that the function g is additive in the sense $g(i_0, i_2) = g(i_0, i_1) + g(i_1, i_2)$. For brevity, we write $g(n)$, $e(n)$ and $f(n)$ to denote $g(0, n)$, $e(0, n)$ and $f(0, n)$, respectively. We can think about the value $g(i, n)$ as of the amount of *energy* we have at the moment i. For each correct transmission, we spend $\log \beta > 0$, for an incorrect one, we get $-\log \alpha > 0$.

8.A.3 High-Level Analysis

The following statement claims that if the amount of energy is sufficiently large, then at some moment the length of the true segment is bounded away from zero irrespective of the actual blocklength n and we still have energy left to further increase the length of the true segment in the remaining steps.

Lemma 8.4. *For every $\varepsilon_1 > \varepsilon_2 > 0$ there exists $\delta > 0$ such that for all but finitely many $\alpha \in (0,1)$ it holds: For $n \in \mathbb{N}$, the condition $g(0,n) > \varepsilon_1 n - \log t(0)$ implies that there is a first moment n_1 such that $t(n_1) \geq \delta$. Furthermore, $g(0,n_1) < \varepsilon_2 n - \log t(0)$ and $g(n_1,n) > (\varepsilon_1 - \varepsilon_2)n$.*

The following lemma shows that after spending enough energy the true segment will become central.

Lemma 8.5. *Let α and β satisfy (8.4). Then there exists a constant $c > 0$ depending only on α, β and $t(n_1)$ such that if $g(n_1,n) > c$, it follows that $T(n)$ is central. Moreover, the length of the true segment at moment n is $1 - o(1)$ as $c \to \infty$.*

Lemmas 8.4 and 8.5 are proved in Section 8.A.4. Finally, we combine these two lemmas to obtain the main result.

Theorem 8.2. *Let the maximal fraction of substitution errors be $\tau \in [0, 1/3)$. For any $\varepsilon > 0$, there exists $\alpha \in [0, (3-\sqrt{5})/2]$ such that for n sufficiently large, the encoding procedure over the adversarial substitution channel with feedback can correct up to τn errors with transmission rate at least $R_s(\tau) - \varepsilon$, where*

$$R_s(\tau) = \begin{cases} 1 - h(\tau), & \text{for } 0 \leq \tau \leq (3-\sqrt{5})/4, \\ (1-3\tau)\log\left(\frac{1+\sqrt{5}}{2}\right), & \text{for } (3-\sqrt{5})/4 < \tau \leq 1/3 \ . \end{cases}$$

Proof. We choose M and accordingly $t(0) = 1/M$ such that $2^{(R_s(\tau)-\varepsilon)n} < M < 2^{(R_s(\tau)-\varepsilon/2)n}$ and $-\log t(0) < (R_s(\tau) - \varepsilon/2)n$. By definition, we have that $e(0,n) \leq \tau n$ and $f(0,n) \geq (1-\tau)n$. For given α and β, we can compute

$$g(0,n) = e(0,n)\log\alpha + f(0,n)\log\beta \geq (\tau\log\alpha + (1-\tau)\log\beta)n\,.$$

Subject to the constraints $\alpha + \beta = 2$ and $\alpha\beta^2 \leq 1$, the function $\tau\log\alpha + (1-\tau)\log\beta$ reaches the maximum $R_s(\tau)$ when $\alpha = 2\min(\tau, (3-\sqrt{5})/4)$ and $\beta = 2 - \alpha$. To make use of Lemma 8.4, we take α slightly smaller than $2\min(\tau, (3-\sqrt{5})/4)$ to have

$$g(0,n) = e(0,n)\log\alpha + f(0,n)\log\beta \geq \left(R_s(\tau) - \frac{\varepsilon}{10}\right)n > \frac{2}{5}\varepsilon n - \log t(0)\,.$$

By applying Lemma 8.4 with $\varepsilon_1 = 2\varepsilon/5$ and $\varepsilon_2 = \varepsilon/5$, there exists a constant $\delta > 0$ such that for all but finite $\alpha \in (0,1)$, there is a first moment n_1 satisfying $t(n_1) \geq \delta$

and furthermore

$$g(n_1, n) = g(0, n) - g(0, n_1) = e(n_1, n) \log \alpha + f(n_1, n) \log \beta > \frac{1}{5} \varepsilon n \,.$$

Thus, for n sufficiently large, we conclude by Lemma 8.5 that $T(n)$ is central. ■

The key ingredient in the proof of Theorem 8.1 is Lemma 8.5. In order to prove it we introduce several auxiliary functions

$$\begin{aligned}
u_1(i) &:= \log(t(i)/x(i)) \\
v_1(i) &:= \log(2t(i)) \\
u_2(i) &:= \begin{cases} -\log(4x(i)y(i)) & \text{if } y(i) \le \frac{1}{2} \\ \log((1-y(i))/x(i)) & \text{if } y(i) > \frac{1}{2} \end{cases} \\
v_2(i) &:= \begin{cases} -\log(2y(i)) & \text{if } y(i) \le \frac{1}{2} \\ \log(2(1-y(i))) & \text{if } y(i) > \frac{1}{2} \end{cases} .
\end{aligned}$$

We also define the constants $u_1' := \log(\beta - \alpha)$, $u_1'' := \log(\beta/\alpha - 1)$, $u_2' := \log(\beta/\alpha)$. Clearly, $u_1' < u_1''$. Finally, define

$$\begin{aligned}
u(i) &:= \begin{cases} u_1(i) & \text{if } u_1(i) < u_1' \\ u_2(i) & \text{if } u_1(i) \ge u_1' \end{cases} \\
v(i) &:= \begin{cases} v_1(i) & \text{if } u_1(i) < u_1'' \\ v_2(i) & \text{if } u_1(i) \ge u_1'' \,. \end{cases}
\end{aligned}$$

To show the validity of Lemma 8.5, we prove that the function v takes a large value at the last moment n, which happens if and only if $l(n)$ and $r(n)$ are small (c.f. Example 8.2).

Proposition 8.2. *If $T(i-1)$ is central and there is a crossing between moments $(i-1)$ and i, then $v(i) - v(i-1) \ge -\log \beta$. Otherwise,*

$$v(i) - v(i-1) \ge \begin{cases} \log \beta & \textit{if transmission is correct} \\ \log \alpha & \textit{if transmission is incorrect}. \end{cases}$$

The change $\Delta v(i) := v(i) - v(i-1)$ of the function v is equal to $g(i-1, i)$, except for the event that $T(i-1)$ is central and there is a crossing. So, if this event is only happening rarely, then $v(n) - v(n_1)$ is close to $g(n_1, n)$. Thus, because $v(n_1)$ can be lower bounded by a function of $t(n_1)$, $v(n)$ is large if $g(n_1, n)$ is large and we are done. To handle the case when those events appear more frequently, we use the function u and the following proposition.

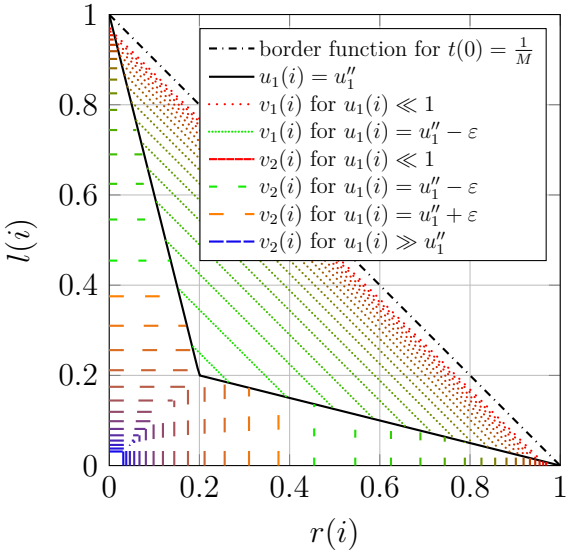

Figure 8.A.3: Illustration for level sets of the function $v(i)$ for $\tau = 0.15$, $\alpha = 2\tau$, and $\beta = 2-\alpha$.

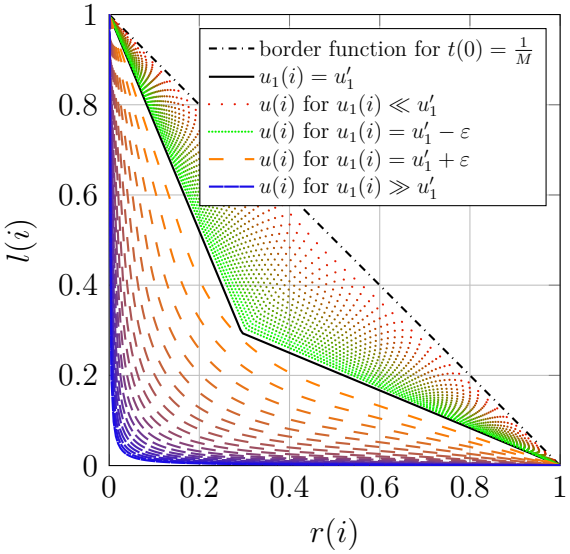

Figure 8.A.4: Illustration for level sets of the function $u(i)$ for $\tau = 0.15$, $\alpha = 2\tau$, and $\beta = 2-\alpha$.

Proposition 8.3. *If $T(i-1)$ is central and there is a crossing, then $u(i) - u(i-1) \geq \log \beta$. Otherwise, $u(i) - u(i-1) \geq 0$.*

So, in this case the function u is big, which implies that the function u_2 is also big. However, this is not enough to conclude that $T(n)$ is central (c.f. Example 8.2). Consider the last occurrence of a crossing with the true segment having been central before the crossing. At this moment, the true segment can be shown to be also balanced, which means that the function v_2 is close to u_2, i.e., it is also big. To make v_2 small again, we must have a lot of incorrect transmissions due to Proposition 8.4.

Proposition 8.4. *If there is a crossing and $u_1(i-1) \geq u_1''$, then $v_2(i) - v_2(i-1) \geq -\log \beta$. If there is no crossing, then*

$$v_2(i) - v_2(i-1) \geq \begin{cases} \log \beta & \text{if the transmission is correct} \\ \log \alpha & \text{if the transmission is incorrect and } v_2(i-1) < \log \beta \\ -\log \beta & \text{if the transmission is incorrect and } v_2(i-1) \geq \log \beta\,. \end{cases}$$

Plenty of incorrect transmissions implies that we have a lot of energy left. Now we can use again Proposition 8.2, since it is not possible that a crossing occurs when the true segment is central and thus, conclude that the function v takes a large value after n transmissions.

Example 8.2. *Figures 8.A.3 and 8.A.4 illustrate different level sets of the functions $v(i)$ and $u(i)$ if they are treated as function on $l(i)$ and $r(i)$. A level set of $v(i)$ is a*

set of pairs $(l(i), r(i))$ such that $v(i) = const$. Note that we use the same color for the same level sets in the graphs. For high level sets of the function $v(i)$, we are close to the origin of the graph, which corresponds to $l(i)$ and $r(i)$ being small and the $t(i)$ very large. In contrast, for high level sets of the function $u(i)$, we can only claim that we are close to an axis, i.e., either $l(i)$ or $r(i)$ is small. Notice that in case $T(i)$ is central, we need to have $l(i), r(i) < \frac{1}{2}$. Combining the two figures for the level sets of $v(i)$ and $u(i)$, it becomes clear that we need to prove that the function $v(n)$ is large in order to conclude that after n transmissions $T(n)$ is central.

Many technical details were hidden in this discussion. In order to give a formal proof of Lemma 8.5, we need some additional propositions.

Proposition 8.5. *We have the following (trivial) properties.*

1. *If there is a crossing, then the transmission is correct.*
2. *If the transmission is correct, then the length of the true segment is increased, i.e., $t(i) > t(i-1)$, otherwise it is decreased.*
3. *If $T(i-1)$ is central, $T(i)$ is not central and there is no crossing, then it follows that the transmission was incorrect.*
4. *The following inequalities always hold $\alpha t(i-1) \leq t(i) \leq \beta t(i-1)$.*
5. *For any $x(i)$, $y(i)$ and $t(i)$, inequalities $u_2(i) \geq u_1(i)$ and $u_2(i) \geq u(i)$ hold.*
6. *For any $x(i)$, $y(i)$ and $t(i)$, inequalities $v_2(i) \geq v_1(i)$ and $v_2(i) \geq v(i)$ hold.*

Proposition 8.6. *We have the following properties.*

1. *If $T(i)$ is central, then $u_2(i) \geq 2v_2(i)$.*
2. *If $T(i)$ is central and balanced, then $u_2(i) < 2v_2(i) + \log(\beta/\alpha)$.*
3. *If there is a crossing between moment $i-1$ and moment i, then $T(i-1)$ and $T(i)$ are balanced.*
4. *If $u_1(i-1) < u_1''$ (or $u_1(i-1) < u_1'$) and $T(i-1)$ is central, and the transmission is correct, then there is a crossing between moment $i-1$ and moment i.*
5. *If $u_1(i-1) \geq u_1'$ (or $u_1(i-1) \geq u_1''$) and there is a crossing between moment $i-1$ and moment i, then $T(i)$ is central.*
6. *If $u_1(i-1) \geq u_1''$ and there is a crossing between moment $i-1$ and i, then $T(i-1)$ is central and $T(i)$ is central.*
7. *If $u_2(i) \geq u_2'$ and there is either a crossing between moment $i-1$ and moment i, or a crossing between moment i and moment $i+1$, then $T(i)$ is central.*

Proposition 8.7. *If $u_1(i-1) < u_1'$, then $u_1(i) - u_1(i-1) \geq 0$. If in addition $T(i-1)$ is central and there is a crossing, then $u_1(i) - u_1(i-1) \geq \log \beta$. On the other hand, if $u_1(i-1) \geq u_1'$, then $u_1(i) \geq u_1'$, and if $u_1(i-1) \geq u_1''$, then $u_1(i) \geq u_1''$.*

Proposition 8.8. *If $u_1(i-1) \geq u_1'$, then $u_2(i) - u_2(i-1) \geq 0$. If both $T(i-1)$ and $T(i)$ are central, then $u_2(i) - u_2(i-1) = -\log(\alpha\beta)$.*

Proposition 8.9. *Suppose $u_1(i-1) < u_1''$. If there is a crossing, then $v_1(i) - v_1(i-1) > 0$. Otherwise,*

$$v_1(i) - v_1(i-1) \geq \begin{cases} \log \beta & \text{if the transmission is correct} \\ \log \alpha & \text{if the transmission is incorrect} \end{cases}.$$

Proposition 8.10. *There exists a constant $c = -5\log(\alpha\beta)$ such that if both $T(i_0)$ and $T(i_1)$ are central and balanced, and $u_2(i_0) \geq c$ then*

$$g(i_0, i_1) \leq u_2(i_1) - u_2(i_0)\,. \tag{8.5}$$

The proof of lemmas and proposition are given in Section 8.A.4 and 8.A.5, respectively. At last, we provide a flowchart in Figure 8.A.5 to make it easier to understand the scheme of the proof.

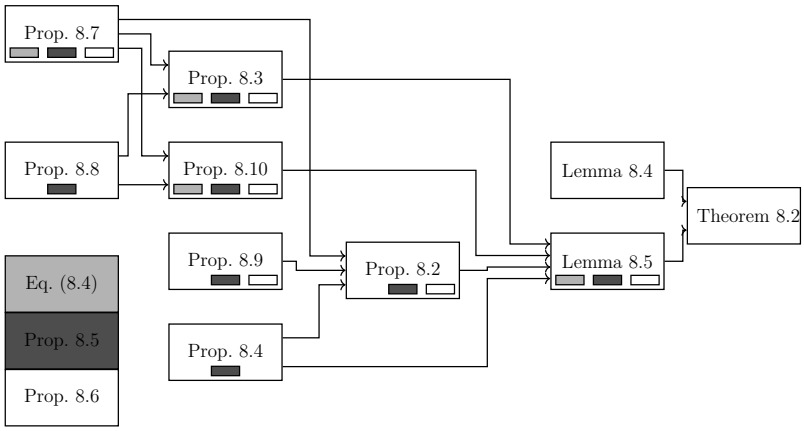

Figure 8.A.5: Illustration of the proof strategy for Theorem 8.2. For the inequality (8.4), Proposition 8.5, or Proposition 8.6 the arrows are omitted due to the sake of clarity. The small indicator boxes however show where they are used.

In prior of the technical proofs of the propositions, we provide in Example 8.3 an illustration for an example transmission, where we track the length of the true segment $T(i)$ and its position.

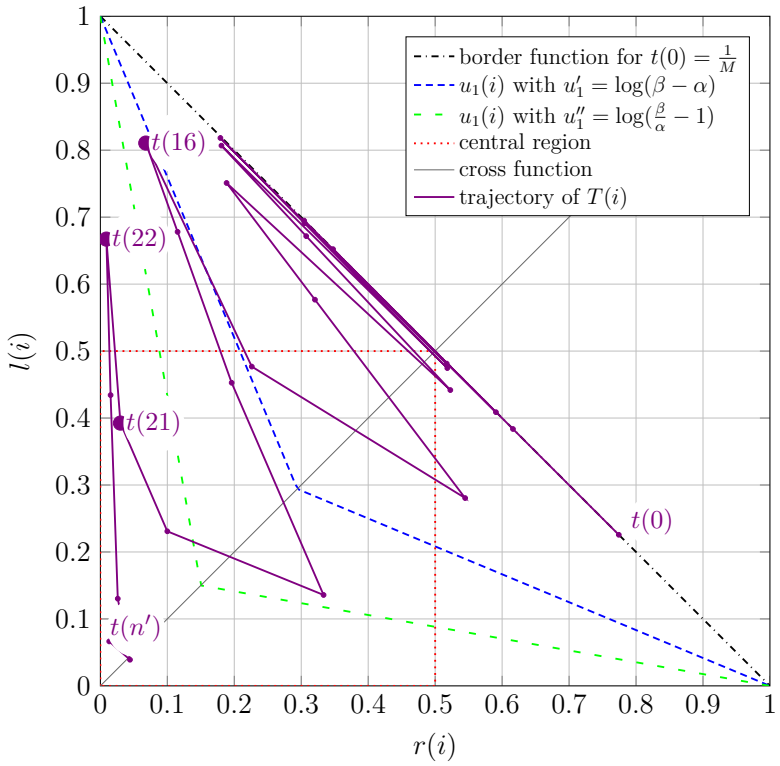

Figure 8.A.6: Illustration for a trajectory of $t(i) = 1 - l(i) - r(i)$. We have $M = 2^{14}$ and transmit a random message, where we use $n' = 27$ total transmissions. Errors occur at time steps 16, 21, and 22. Used parameters are $\tau = 0.15$, $\alpha = 2\tau$, and $\beta = 2 - \alpha$.

Example 8.3. *We illustrate in Figure 8.A.6 an example transmission of a random message m out of $M = 2^{14}$ for $n' = 27$ total transmission. We use that $\tau = 0.15$, $\alpha = 2\tau$ and $\beta = 2 - \alpha$. Recall that $t(i) = 1 - l(i) - r(i)$. We have three bit-flip errors at the time steps $i \in \{16, 21, 22\}$. Since at the first transmission all segments are the same, thus the position of the true segment $T(0)$ described by the pair $(l(0), r(0))$ lies close to the line $l(i) + r(i) = 1$. For more and more bit transmissions, we see that the trajectory of $T(i)$ is getting closer to the origin, meaning that its length is increasing. We have that $T(i)$ is central at time step $i = 15$ after multiple crosses before. Since at time step $i = 16$ an error occurred, the trajectory leaves the central region. Furthermore, at the same time step, the trajectory crosscuts the level set u_1' and does not cross it again in later time steps (c.f. Proposition 8.7). The same behavior we observe if the level set u_1'' is crosscut although we have errors at time steps $i = 21, 22$, which makes $T(i)$ not central anymore (c.f. Proposition 8.7). For the remaining correct transmissions, we see that $t(i)$ is always increasing until we reach the last n' transmission.*

8.A.4 Proofs of Lemmas

Proof of Lemma 8.4. We choose $k \in \mathbb{N}$ such that $k > \frac{\log \beta}{\varepsilon_2}$ for $\varepsilon_1 > \varepsilon_2 > 0$. Let us fix $\delta < \frac{1}{\beta^2}$ which corresponds to the smallest displacement of the position $\frac{1}{2}$ after $1, 2, \ldots, k-1$ transmissions. For instance, after transmitting one symbol the point $\frac{1}{2}$ can be mapped to one of the points $\{\frac{\alpha}{2}, \frac{\beta}{2}\}$, whereas after two transmissions the image of the point $\frac{1}{2}$ is one of the points $\{\frac{\alpha^2}{2}, \frac{\alpha\beta}{2}, 1-\alpha(1-\frac{\beta}{2}), 1-\beta(1-\frac{\beta}{2})\}$. One can prove that $\delta > 0$ for all but finite $\alpha \in (0,1)$. Remark that if α is a non-algebraic number, then $\delta > 0$ for any k.

Assume on the contrary that $t(i) < \delta$ for $i \in \{0, 1, \ldots, n\}$. We have by the definition of δ that if $T(i)$ is central, i.e., the point $\frac{1}{2}$ is in $T(i)$, then $T(i+1), \ldots, T(i+k-1)$ are not central. Let us denote as n' the number of moments when $T(i)$ is central. By the definition we know that $n' \leq \frac{n}{k} + 1 < \frac{\varepsilon_1 n}{\log \beta} + 1$. Therefore, we conclude

$$t(n) \geq t(0)\alpha^{e(0,n)}\beta^{f(0,n)-n'} \geq t(0)\alpha^{e(0,n)}\beta^{f(0,n)-\frac{\varepsilon_1 n}{\log\beta}-1} \geq \frac{1}{\beta} > \delta\,,$$

where we used the fact that $g(0,n) = e(0,n)\log\alpha + f(0,n)\log\beta > \varepsilon_1 n - \log t(0)$. Hence, we come to a contradiction.

Let n_1 be a first moment when $t(n_1) \geq \delta$. Applying the same strategy we can see that

$$\beta\delta > \beta t(n_1 - 1) \geq t(n_1) \geq t(0)\alpha^{e(0,n_1)}\beta^{f(0,n_1)-\frac{\varepsilon_2 n_1}{\log\beta}-1}\,.$$

From the two aforementioned observations it follows that

$$g(0,n_1) = e(0,n_1)\log\alpha + f(0,n_1)\log\beta < \varepsilon_2 n_1 - \log t(0) + \log(\beta^2\delta) < \varepsilon_2 n - \log t(0)\,.$$

Hence

$$g(n_1, n) = g(0,n) - g(0,n_1) > (\varepsilon_1 - \varepsilon_2)n > 0\,.$$

■

Proof of Lemma 8.5. Set $I := \{i \in (n_1, n] : T(i-1)$ is central and there is a crossing between moments $i-1$ and $i\}$. Let $m := |I|$, and take $c \gg m' \gg v_2' \gg 1$. One way to think about the choices of the parameters is to let $c = x^3, m' = x^2, v_2' = x$, where x is sufficiently large relative to $\alpha, \beta, t(n_1)$.

Case $m \leq m'$. From Proposition 8.5.6 and the definition of v it follows that $v(n_1) \geq v_1(n_1) = \log(2t(n_1))$. Thus, by Propositions 8.2, we have

$$\begin{aligned} v(n) &\geq v(n_1) + e(n_1,n)\log\alpha + (f(n_1,n) - m)\log\beta - m\log\beta \\ &\geq \log(2t(n_1)) + g(n_1,n) - 2m'\log\beta \\ &> \log(2t(n_1)) + c - 2m'\log\beta \gg 1\,. \end{aligned}$$

Here we used our choice of c and m' and the fact that $g(n_1,n) = e(n_1,n)\log\alpha +$

$f(n_1, n) \log \beta$. From the definition of function v and the fact that $v(n) \gg 1$ we conclude that $T(n)$ is central (see Fig. 8.A.3).

Case $m > m'$. Let n_2 be the m'th smallest element of I. From Proposition 8.5.5 and the definition of function u it follows that $u_2(n_2) \geq u(n_2)$, and $u(n_1) \geq u_1(n_1) = \log(t(n_1)/x(n_1))) \geq \log(2t(n_1))$ as $x(n_1) \leq 1/2$. Combining this together with Proposition 8.3 we get

$$u_2(n_2) \geq u(n_2) \geq u(n_1) + m' \log \beta \geq \log(2t(n_1)) + m' \log \beta = \Omega(m') \,.$$

The last equality holds due to our choice of m'. Thus, by Proposition 8.6.7, $T(n_2)$ is central and by Proposition 8.6.3, $T(n_2)$ is balanced. By Propositions 8.2, 8.5.6 and 8.6.1, we know that

$$\begin{aligned} g(n_1, n_2) &\leq v(n_2) - v(n_1) + 2m' \log \beta \leq v_2(n_2) - v_1(n_1) + 2m' \log \beta \\ &\leq \frac{1}{2} u_2(n_2) - \log(2t(n_1)) + 2m' \log \beta \leq \frac{1}{2} u_2(n_2) + O(m') \,. \end{aligned} \tag{8.6}$$

Let n_3 be the last element of I. We know that $T(n_2)$ is central and balanced, and $u_2(n_2) = \Omega(m')$. The same conclusion holds for $T(n_3)$, in particular, $T(n_3)$ is central and balanced, and $u_2(n_3) = \Omega(m')$. By Proposition 8.10, we obtain

$$g(n_2, n_3) \leq u_2(n_3) - u_2(n_2) \,. \tag{8.7}$$

By Proposition 8.6.2, we derive that

$$v_2(n_3) \geq \frac{1}{2} u_2(n_3) - \frac{1}{2} \log(\beta/\alpha) = \Omega(m') \gg v_2' \,.$$

If $v_2(n) \geq v_2'$, we are done. Otherwise, let $n_4 > n_3$ be a first moment when $v_2(n_4) < v_2'$. Since $v_2(i) > 0$ for all $i \in [n_3, n_4)$, from the definition of the function v_2 it follows that $T(i)$ is central for all $i \in [n_3, n_4)$. By the definition of I, there are no crossings between n_3 and n_4. By Propositions 8.4 and 8.6.2 and the fact $\alpha\beta < 1$, we obtain

$$\begin{aligned} g(n_3, n_4) &= e(n_3, n_4) \log \alpha + f(n_3, n_4) \log \beta \leq 2(-e(n_3, n_4) + f(n_3, n_4)) \log \beta \\ &\leq 2v_2(n_4) - 2v_2(n_3) < -u_2(n_3) + O(v_2') \,. \end{aligned} \tag{8.8}$$

By the definition of I, there is no $i \in (n_4, n]$ such that $T(i-1)$ is central and there is a crossing between moments $i-1$ and i. By Proposition 8.2, we have

$$g(n_4, n) \leq v(n) - v(n_4) \,. \tag{8.9}$$

Since there is no crossing between $n_4 - 1$ and n_4, by Proposition 8.4, we have $v_2(n_4) \geq v_2(n_4 - 1) - \log \beta \geq v_2' - \log \beta \gg 1$. By the definition of $v_2(n_4)$, we know that $y(n_4) \ll 1$ and so $v_1(n_4) > 0$, hence $v(n_4) \geq 0$. Adding up (8.6)-(8.9) and the fact that $v(n_4) \geq 0$

yield

$$v(n) \geq g(n_1, n) - O(m') - O(v_2') \gg 1\,,$$

which implies that $T(n)$ is central. ■

8.A.5 Proofs of Propositions

Proof of Proposition 8.5. 1: If there is a crossing, then the smaller segment among $\{L(i), R(i)\}$ has been increased. It happens only when the transmission is correct.
2: We only proof the assertion for correct transmission because the case for incorrect transmission is proved analogously. In case that $T(i-1)$ is non-central the statement follows because $t(i) = t(i-1)\beta$. In case $T(i-1)$ is central we have that

$$t(i) = 1 - \beta x(i-1) - \alpha y(i-1)$$

and therefore

$$t(i) - t(i-1) = (1-\beta)x(i-1) + (1-\alpha)y(i-1) = \frac{\beta - \alpha}{2}(y(i-1) - x(i-1)) > 0\,,$$

where we have used that $\alpha + \beta = 2$.
3: We assume that the transmission is correct and there is no crossing. Then since $T(i-1)$ is central it holds that

$$\begin{aligned} x(i-1)\beta &= x(i)\,, \\ y(i-1)\alpha &= y(i)\,. \end{aligned} \tag{8.10}$$

In order for $T(i)$ to be non central it is necessary that $y(i) > \frac{1}{2}$ but this leads to a contradiction due to (8.10) and because $\alpha < 1$ and $y(i-1) < \frac{1}{2}$.
4: If $T(i-1)$ is not central its length is multiplied by β for a correct transmission and by α for an erroneous one. Because the length of all subsegments combined remains one during the entire process, the length of the true segment is multiplied by a value in $[\alpha, \beta]$ if $T(i-1)$ is central, completing the proof.
5: Recall that

$$u_2(i) := \begin{cases} -\log(4x(i)y(i)) & \text{if } y(i) \leq \frac{1}{2} \\ \log((1-y(i))/x(i)) & \text{if } y(i) > \frac{1}{2}\,. \end{cases}$$

We claim that

$$-\log(4x(i)y(i)) \geq \log((1-y(i))/x(i))\,.$$

The following statements are equivalent.

$$\frac{1}{4x(i)y(i)} \geq \frac{1-y(i)}{x(i)}\,,$$
$$1 \geq 4y(i)(1-y(i))\,.$$

The last inequality follows because its right hand side is maximized for $y(i) = \frac{1}{2}$. This maximal value is equal to one and our claim follows. It follows that $u_2(i) \geq u_1(i) = \log(t(i)/x(i))$ because

$$\frac{1-y(i)}{x(i)} \geq \frac{t(i)}{x(i)} = 2^{u_1(i)}\,.$$

6: Recall that

$$v_2(i) := \begin{cases} -\log(2y(i)) & \text{if } y(i) \leq \frac{1}{2} \\ \log(2(1-y(i))) & \text{if } y(i) > \frac{1}{2} \end{cases}\,.$$

Similar to the proof of Proposition 8.5.5 we show first that

$$-\log(2y(i)) \geq \log(2(1-y(i)))\,. \tag{8.11}$$

The following statements are equivalent.

$$\frac{1}{2y(i)} \geq 2(1-y(i))\,,$$
$$1 \geq 4y(i)(1-y(i))\,,$$

where the last inequality follows again because its right hand side is maximized for $y(i) = \frac{1}{2}$. This maximal value is again equal to one and we have shown inequality (8.11). It follows that $v_2(i) \geq v_1(i) = \log(2t(i))$ because

$$\log(2(1-y(i)) \geq \log(2(1-x(i)-y(i))) = v_1(i)\,.$$

■

Proof of Proposition 8.6. 1: Since $T(i)$ is central, we have $y(i) \leq \frac{1}{2}$. Thus,

$$u_2(i) = -\log(4x(i)y(i)) \geq -\log(4y^2(i)) = 2v_2(i)\,.$$

2: Since $T(i)$ is central and balanced, we have $y(i) \leq \frac{1}{2}$ and $y(i)/x(i) \leq \beta/\alpha$. Hence,

$$\begin{aligned} u_2(i) &= -\log(4x(i)y(i)) \\ &= -\log(4y^2(i)x(i)/y(i)) \\ &= -\log(4y^2(i)) + \log(y(i)/x(i)) \\ &\leq 2v_2(i) + \log(\beta/\alpha)\,. \end{aligned}$$

3: Without loss of generality, assume that $l(i-1) \leq \frac{1}{2} \leq r(i-1)$ and $l(i) > r(i)$. From Proposition 8.5.1 it follows that $l(i) = \beta l(i-1)$ and $r(i) \geq \alpha r(i-1)$. Then

$$\frac{y(i)}{x(i)} = \frac{l(i)}{r(i)} \leq \frac{\beta l(i-1)}{\alpha r(i-1)} \leq \frac{\beta r(i-1)}{\alpha r(i-1)} = \frac{\beta}{\alpha}\,.$$

This means that $T(i)$ is balanced. Similarly, we check that $T(i-1)$ is balanced

$$\frac{y(i-1)}{x(i-1)} = \frac{r(i-1)}{l(i-1)} \leq \frac{r(i)/\alpha}{l(i)/\beta} < \frac{\beta l(i)}{\alpha l(i)} = \frac{\beta}{\alpha}\,.$$

4: First we note that $u_1' < u_1''$. Without loss of generality, assume that $l(i-1) < r(i-1) \leq \frac{1}{2}$. From the condition $u_1(i-1) \leq u_1''$ it follows that

$$\frac{t(i-1)}{x(i-1)} = \frac{1-l(i-1)-r(i-1)}{l(i-1)} = \frac{1-r(i-1)}{l(i-1)} - 1 \leq \frac{\beta}{\alpha} - 1\,.$$

This implies $\beta l(i-1)+\alpha r(i-1) \geq \alpha$ or $\beta l(i-1) \geq \alpha - \alpha r(i-1)$. Since the transmission is correct, we have $l(i) = \beta l(i-1)$. Thus, $l(i) \geq \alpha - \alpha r(i-1)$. Since $r(i-1) \leq \frac{1}{2}$, we have $l(i) > \alpha/2 > \alpha r(i-1) = r(i)$. This proves that there is a crossing between moment $i-1$ and moment i.
5: Without loss of generality, we assume that $l(i-1) \leq r(i-1)$. From the condition $u_1(i-1) \geq u_1'$, we derive

$$\frac{1-2l(i-1)}{l(i-1)} \geq \frac{1-l(i-1)-r(i-1)}{l(i-1)} = \frac{t(i-1)}{x(i-1)} \geq 2^{u_1'} = \beta - \alpha = 2\beta - 2$$

and $l(i-1) \leq 1/(2\beta)$. Since there is a crossing, we have $\beta l(i-1) = l(i) \geq r(i)$ and hence $y(i) = l(i) \leq \frac{1}{2}$.
6: Without loss of generality, assume that $l(i-1) < r(i-1)$ and $l(i) \geq r(i)$. First we note that a crossing may happen only when the transmission is correct. From condition $u_1(i-1) \geq u_1''$ we derive that

$$\frac{t(i-1)}{x(i-1)} = \frac{1-l(i-1)-r(i-1)}{l(i-1)} = \frac{1-r(i-1)}{l(i-1)} - 1 \geq \frac{\beta}{\alpha} - 1\,.$$

This yields that $\beta l(i-1) + \alpha r(i-1) \leq \alpha$. Since the transmission is correct, we have $l(i) = \beta l(i-1)$, and, thus, $l(i) + \alpha r(i-1) \leq \alpha$ or $r(i-1) \leq 1 - l(i)/\alpha$. Because of the property $l(i) \geq r(i)$, we get $r(i-1) \leq 1 - r(i)/\alpha$. Since $r(i) \geq r(i-1)\alpha$, we obtain $r(i-1) \leq 1 - r(i-1)$, which is equivalent to that $T(i-1)$ is central. To show that $T(i)$ is central, observe that $u_1(i-1) \geq u_1'' \geq u_1'$ and we can use Proposition 8.6.5.
7: Suppose there is either a crossing between moment $i-1$ and moment i or a crossing between moment i and moment $i+1$. Without loss of generality, assume that $l(i) \geq r(i)$ and either $l(i-1) < r(i-1)$ or $l(i+1) < r(i+1)$. Toward a contradiction, assume that

$T(i)$ is not central and, thus, $y(i) = l(i) > \frac{1}{2}$. The condition $u_2(i) \geq u_2'$ is equivalent to that

$$\frac{1 - y(i)}{x(i)} = \frac{1 - l(i)}{r(i)} \geq \frac{\beta}{\alpha}\,.$$

This implies $\alpha l(i) + \beta r(i) \leq \alpha$. Since $l(i) > \frac{1}{2}$, we have either $r(i) \geq \alpha r(i-1) > \alpha l(i-1) = \alpha l(i)/\beta > \alpha/(2\beta)$ or $\beta r(i) = r(i+1) > l(i+1) \geq l(i)\alpha > \alpha/2$. Thus, we obtain $\alpha l(i) + \beta r(i) > \alpha/2 + \alpha/2 = \alpha$. This contradicts our assumption and, hence, $T(i)$ is central. ■

Proof of Proposition 8.7. We consider first the part of the proposition where the condition $u_1(i-1) < u_1'$ holds. There are several cases that need to be distinguished here. We first consider the case when an error occurred. Then $t(i) \geq \alpha t(i-1)$ (Proposition 8.5.4) and $x(i) = \alpha x(i-1)$ and thus, $u_1(i) = \log(t(i)/x(i)) \geq \log(t(i-1)/x(i-1)) = u_1(i-1)$. Next we proceed to the case that the transmission was correct. If there is no crossing, then by Proposition 8.6.4 we have that $T(i-1)$ has to be non-central and, thus, $t(i) = \beta t(i-1)$ and $x(i) = \beta x(i-1)$, which implies $u_1(i) = u_1(i-1)$.

Now suppose that there is a crossing which implies that no error occurred. We consider the case that $T(i-1)$ is not central. Because we have a crossing we know that $x(i) \leq \beta x(i-1)$ and therefore

$$2^{u_1(i)} = \frac{t(i)}{x(i)} \geq \frac{\beta t(i-1)}{\beta x(i-1)} = \frac{t(i-1)}{x(i-1)} = 2^{u_1(i-1)}\,.$$

It remains to check the case when $T(i-1)$ is central. By the statement we need to show

$$\begin{aligned} u_1(i) - u_1(i-1) &= \log\left(\frac{1 - x(i-1)\beta - y(i-1)\alpha}{y(i-1)\alpha}\right) - \log\left(\frac{1 - x(i-1) - y(i-1)}{x(i-1)}\right) \\ &\geq \log\beta\,. \end{aligned}$$

This is equivalent to

$$\begin{aligned} &(1 - x(i-1)\beta - y(i-1)\alpha)x(i-1) \geq (1 - x(i-1) - y(i-1))y(i-1)\alpha\beta\,, \\ &x(i-1)\frac{\beta - \alpha}{2}(y(i-1) - x(i-1)) \geq (1 - x(i-1) - y(i-1)(y(i-1)\alpha\beta - x(i-1))\,. \end{aligned}$$

As the left hand side of this equation is always positive, we only have to consider the case where the right hand side has this property as well. Since $u_1(i-1) < u_1'$ we need prove that

$$\begin{aligned} y(i-1) - x(i-1) &> 2(y(i-1)\alpha\beta - x(i-1))\,, \\ x(i-1) &> y(i-1)(2\alpha\beta - 1)\,. \end{aligned}$$

Because there is a crossing we know from Proposition 8.6.3 that

$$x(i-1) > y(i-1)\frac{\alpha}{\beta}\,.$$

To show $2\alpha\beta - 1 < \frac{\alpha}{\beta}$ we use the chain of equivalent inequalities

$$\begin{aligned} 2\alpha\beta - 1 &< \frac{\alpha}{\beta}\,, \\ 2\alpha\beta^2 - \beta &< \alpha = 2 - \beta\,, \\ \alpha\beta^2 &< 1\,, \end{aligned}$$

where the last line follows from (8.4). This completes the first part of the proposition.

Now we assume that $u_1(i-1) \geq u_1'$ and and show that in this case $u_1(i) \geq u_1'$. We assume that the transmission was incorrect. Then

$$\frac{t(i)}{x(i)} \geq \frac{\alpha t(i-1)}{\alpha x(i-1)} = \frac{t(i-1)}{x(i-1)}\,.$$

Therefore, from now on we only consider successful transmissions. First, we have a look at the case that $T(i-1)$ is not central. Then

$$\frac{t(i)}{x(i)} \geq \frac{t(i-1)\beta}{x(i-1)\beta} = \frac{t(i-1)}{x(i-1)}\,,$$

where the inequality followed because $\beta x(i-1) \geq x(i)$ with equality if there is no crossing and a strict inequality if there is a crossing. Therefore, we have shown the assertion for both cases.

Next we analyze the case where $T(i-1)$ is central and there is no crossing

$$\frac{t(i)}{x(i)} \geq \frac{t(i-1)}{\beta x(i-1)} \geq \frac{\beta - \alpha}{\alpha\beta} > \beta - \alpha\,.$$

The first inequality follows because correct transmission can never reduce the length of the true segment. The second one follows because otherwise by Proposition 8.6.4 there would be a crossing and the last inequality follows because $\alpha\beta < 1$.

Next we consider the case that $T(i-1)$ is central and there is a crossing. We observe that since there is a crossing $T(i-1)$ is balanced by Proposition 8.6.3. Therefore, $y(i-1)/x(i-1) < \beta/\alpha$ and

$$\frac{t(i)}{x(i)} = \frac{1 - x(i-1)\beta - y(i-1)\alpha}{y(i-1)\alpha} = \frac{1 - x(i-1)\beta}{y(i-1)\alpha} - 1\,. \tag{8.12}$$

Since $u_1(i-1) \geq u_1'$ we obtain

$$\begin{aligned}
\frac{1-x(i-1)-y(i-1)}{x(i-1)} &\geq \beta-\alpha\,,\\
1-x(i-1)-y(i-1) &\geq (\beta-\alpha)x(i-1)\,,\\
1-x(i-1)-\frac{\beta-\alpha}{2}x(i-1)-y(i-1) &\geq \frac{\beta-\alpha}{2}x(i-1)\,,\\
1-\beta x(i-1)-y(i-1) &\geq \frac{\beta-\alpha}{2}x(i-1)\,.
\end{aligned}$$

Therefore, we get

$$\frac{1-x(i-1)\beta}{y(i-1)\alpha} \geq \frac{\frac{\beta-\alpha}{2}x(i-1)+y(i-1)}{y(i-1)\alpha} = \frac{1}{\alpha}+\frac{x(i-1)\frac{\beta-\alpha}{2}}{y(i-1)\alpha}\,. \tag{8.13}$$

By using the fact that $T(i-1)$ is balanced we get

$$\frac{1}{\alpha}+\frac{x(i-1)\frac{\beta-\alpha}{2}}{y(i-1)\alpha} > \frac{1}{\alpha}+\frac{\beta-\alpha}{2\beta} = \frac{2\beta+\alpha\beta-\alpha^2}{2\alpha\beta} = \frac{(\alpha+\beta)\beta+\alpha\beta+\alpha^2-2\alpha^2}{2\alpha\beta}\,, \tag{8.14}$$

which we simplify to

$$\frac{(\alpha+\beta)^2-2\alpha^2}{2\alpha\beta} = \frac{2-\alpha^2}{\alpha\beta} = 1+\frac{2-2\alpha}{\alpha\beta} = 1+\frac{\beta-\alpha}{\alpha\beta} > \beta-\alpha+1\,, \tag{8.15}$$

where the last inequality follows because $1 > \alpha\beta$. Combining equation (8.12) and the inequalities (8.13), (8.14), (8.15) completes the proof of this part.

Now we proof that if $u_1(i-1) \geq u_1''$, then $u_1(i) \geq u_1''$. Seeking for a contradiction, assume $u_1(i-1) \geq u_1''$ and $u_1(i) < u_1''$, i.e., u_1 is decreasing. The cases that $T(i-1)$ is not central or the transmission was incorrect can be handled in the same manner as in the previous part of the proof. Therefore, $T(i-1)$ is central and the transmission is correct. The following inequalities are equivalent

$$\begin{aligned}
u_1(i-1) &\geq u_1'' = \log(\beta/\alpha-1)\,,\\
\frac{1-x(i-1)-y(i-1)}{x(i-1)} &\geq \beta/\alpha-1\,,\\
\alpha &\geq \alpha y(i-1)+\beta x(i-1)\,.
\end{aligned}$$

In the same manner, we can obtain the equivalent relations

$$\begin{aligned}
u_1(i) &< u_1'',\\
\alpha &< \alpha y(i)+\beta x(i)\,.
\end{aligned}$$

Since $T(i-1)$ is central and the transmission is correct, we conclude that $\alpha y(i-1) + \beta x(i-1) = x(i) + y(i)$. Using this equality results in a contradiction

$$\alpha \geq \alpha y(i-1) + \beta x(i-1) = y(i) + x(i) \geq \alpha y(i) + \beta x(i) > \alpha \,.$$

■

Proof of Proposition 8.8. We are first considering the case that both $T(i-1)$ and $T(i)$ are central. Whether there occurs an error or not it holds that

$$u_2(i) - u_2(i-1) = -\log(4x(i-1)\beta y(i-1)\alpha) + \log(4x(i-1)y(i-1)) = -\log(\alpha\beta) > 0\,,$$

since $\alpha\beta < 1$.

Now we consider the case that $T(i-1)$ is central and $T(i)$ is not central. Due to Proposition 8.6.5 there is no crossing and we know because of Proposition 8.5.3 that an error occurred. It follows that

$$u_2(i) - u_2(i-1) = \log\left(\frac{1-y(i)}{x(i)}\right) + \log(4x(i-1)y(i-1)) = \log\left(\frac{4(1-y(i))y(i)}{\alpha\beta}\right).$$

Because $T(i)$ is not central we have that $\frac{1}{2} \leq y(i) \leq \frac{\beta}{2}$ where $y(i) = \frac{\beta}{2}$ minimizes the last logarithm, leading to

$$\log\left(\frac{4(1-y(i))y(i)}{\alpha\beta}\right) \geq \log 1 = 0\,.$$

Next we are considering the case that $T(i-1)$ is not central and there occurs a crossing. Without loss of generality we consider that $l(i-1) < r(i-1)$. Due to Proposition 8.6.5 we know that $T(i)$ is central and therefore we only have to consider this case. Then, it holds that

$$\begin{aligned} u_2(i) - u_2(i-1) &= -\log(4l(i)r(i)) - \log\left(\frac{1-r(i-1)}{l(i-1)}\right) \\ &= -\log\left(\frac{4(1-r(i-1))l(i)r(i)}{l(i-1)}\right) \\ &= -\log(4\beta(1-r(i-1))r(i))\,. \end{aligned}$$

Next we use that $r(i) = 1 - \beta(t(i-1) + l(i-1)) = \frac{\alpha}{2} + \beta\left(r(i-1) - \frac{1}{2}\right)$ and obtain

$$-\log(4\beta(1-r(i-1))r(i)) = -\log\left(4\beta(1-r(i-1))\left(\frac{\alpha}{2} + \beta\left(r(i-1) - \frac{1}{2}\right)\right)\right).$$

By looking at the quadratic form (in $r(i-1)$) we find its maximum for $r(i-1) = \frac{3\beta-\alpha}{4\beta}$

and

$$-\log\left(4\beta(1-r(i-1))\left(\frac{\alpha}{2}+\beta\left(r(i-1)-\frac{1}{2}\right)\right)\right) \geq -\log\left((\beta+\alpha)\left(\frac{\alpha}{2}+\frac{\beta-\alpha}{4}\right)\right) = -\log 1 = 0\,.$$

It remains to check the case when $T(i-1)$ is not central and there is no crossing. The true segment $T(i)$ can be either central, or non-central. Since $\frac{1}{4x(i)y(i)} \geq \frac{1-y(i)}{x(i)}$, it holds that

$$u_2(i) - u_2(i-1) \geq \log\left(\frac{1-y(i)}{x(i)}\right) - \log\left(\frac{1-y(i-1)}{x(i-1)}\right) = 0\,.$$

The last equality holds since there is no crossing and thus we have

$$1-y(i) = \begin{cases}(1-y(i-1))\beta & \text{for correct transmission}\\ (1-y(i-1))\alpha & \text{for incorrect transmission}\end{cases}$$

and

$$x(i) = \begin{cases}x(i-1)\beta & \text{for correct transmission}\\ x(i-1)\alpha & \text{for incorrect transmission}\,.\end{cases}$$

■

Proof of Proposition 8.9. If there is a crossing, then because the transmission has to be correct due to Proposition 8.5.1. In that case the true segment is enlarged by Proposition 8.5.2 and the first statement follows. For the second statement we first consider the case that $T(i-1)$ is not central. Then it is clear that

$$v_1(i) - v_1(i-1) = \log t(i) - \log t(i-1) = \begin{cases}\log\beta & \text{if the transmission is correct}\\ \log\alpha & \text{if the transmission is incorrect}\,.\end{cases}$$

The other case to be considered is if $T(i-1)$ is central and there is no crossing. In that case the transmission has to be incorrect because otherwise a crossing would occur, according to Proposition 8.6.4. We have to show that

$$\log t(i) - \log t(i-1) \geq \log\alpha\,,$$

which is always true due to Proposition 8.5.4. ■

Proof of Proposition 8.4. We first consider the case that there is a crossing and $u_1(i-1) \geq u_1''$. Due to Proposition 8.6.6 we know that $T(i-1)$ and $T(i)$ are both

central. It follows that

$$\begin{aligned} v_2(i) - v_2(i-1) &= -\log(2y(i)) + \log(2y(i-1)) = -\log(2x(i-1)\beta) + \log(2y(i-1)) \\ &\geq -\log\beta\,. \end{aligned}$$

For the remainder of the proof we are considering cases without a crossing. We consider the case that $T(i-1)$ and $T(i)$ are central. We obtain that

$$\begin{aligned} v_2(i) - v_2(i-1) &= \log\left(\frac{y(i-1)}{y(i)}\right) \\ &= \begin{cases} -\log\alpha \geq \log\beta & \text{if the transmission was correct} \\ -\log\beta \geq \log\alpha & \text{if the transmission was incorrect}\,. \end{cases} \end{aligned}$$

Next we consider the case that $T(i-1)$ is central and $T(i)$ is not central. This can only occur in the incorrect case (Proposition 8.5.3) and then

$$v_2(i) - v_2(i-1) = \log(2(1-y(i))) + \log(2y(i-1)) = \log(4(1-y(i))y(i)) - \log\beta\,,$$

but since $y(i-1) \leq \frac{1}{2}$ we know that $y(i) \leq \frac{\beta}{2}$ and we get

$$v_2(i) - v_2(i-1) = \log(4(1-y(i))y(i)) - \log\beta \geq \log(\alpha\beta) - \log\beta = \log\alpha\,.$$

In the considered case, it also holds that $v_2(i-1) < \log\beta$ since the following statements are equivalent

$$\begin{aligned} v_2(i-1) &< \log\beta\,, \\ -\log(2y(i-1)) &< \log\beta\,, \\ \frac{1}{2} &< \beta y(i-1) = y(i)\,, \end{aligned}$$

where the last inequality holds because $T(i)$ is not central.

Now we consider the case that $T(i-1)$ and $T(i)$ are both not central and there is no crossing. In this case it holds that

$$\begin{aligned} v_2(i) - v_2(i-1) &= \log(2(1-y(i))) - \log(2(1-y(i-1))) = \log\left(\frac{t(i)+x(i)}{t(i-1)+x(i-1)}\right) \\ &= \begin{cases} \log\beta & \text{if the transmission was correct} \\ \log\alpha & \text{if the transmission was incorrect}\,. \end{cases} \end{aligned}$$

We show that $v_2(i-1) < \log\beta$ if the transmission was incorrect. It holds that

$$\log(2(1-y(i-1))) \leq -\log(2y(i-1))\,.$$

It follows that $v_2(i-1) < \log\beta$ because the following statements are equivalent

$$-\log(2y(i-1)) < \log\beta, \quad \frac{1}{2y(i-1)} < \beta, \quad \frac{1}{2} < \beta y(i-1) = y(i)\,,$$

where the last line is true because $T(i)$ is not central and the transmission was incorrect.

Finally we consider the case that $T(i-1)$ is not central but $T(i)$ is central. In this case, the transmission is correct since otherwise it holds $y(i) \geq y(i-1) \geq \frac{1}{2}$. Furthermore, recall that

$$-\log(2y(i)) \geq \log(2(1-y(i)))\,.$$

Therefore, we obtain

$$v_2(i) - v_2(i-1) = -\log(2y(i)) - \log(2(1-y(i-1))) \geq \log\left(\frac{1-y(i)}{1-y(i-1)}\right) = \log\beta\,,$$

which concludes the proof. ■

Proof of Proposition 8.10. From Proposition 8.6.2 and the inequality (8.4) we deduce that

$$v_2(i_0) > \frac{u_2(i_0)}{2} - \frac{\log(\beta/\alpha)}{2} \geq -\frac{1}{2}\log(\alpha^4\beta^6) \geq -\frac{1}{2}\log\beta^{-2} = \log\beta\,. \tag{8.16}$$

Since $v_2(i_0) = -\log(2y(i_0)) > \log\beta$, we have $y(i_0) < 1/(2\beta)$. From $y(i_0) < 1/(2\beta)$ we obtain that

$$u_1(i_0) = \log\left(\frac{1-y(i_0)-x(i_0)}{x(i_0)}\right) \geq \log\left(\frac{1-2y(i_0)}{y(i_0)}\right) > u_1'\,.$$

From Proposition 8.7 it follows that $u_1(i) \geq u_1'$ for all $i \geq i_0$. Proposition 8.8 implies that $u_2(i) \geq c$ for all $i \geq i_0$. Using arguments as in (8.16), we get $v_2(i) > \log\beta$ for all i such that $T(i)$ is central and balanced. Note that since $c \geq u_2'$, the inequality $u_2(i) \geq u_2'$ holds.

Say that there are crossings at the moments $j_1, \ldots, j_k$, $i_0 \leq j_1 < \ldots < j_k < i_1$. Partition $[i_0, i_1)$ into the union $[i_0, j_1) \cup [j_1, j_1+1) \cup [j_1+1, j_2] \cup \ldots \cup [j_k, i_1)$. By Proposition 8.6.3 and 8.6.7 the true segment is central and balanced at all the endpoints of the segments. Thus, all conditions of Proposition 8.10 are satisfied. Moreover, if we prove the inequality (8.5) for these segments, it would also be true for their union. Therefore, it is sufficient to consider only two cases:

1. $i_1 = i_0 + 1$ and there is a crossing.
2. there is no crossing between i_0 and i_1.

The first case is trivial. There is a crossing between i_0 and $i_1 = i_0 + 1$, hence the transmission is correct by Proposition 8.5.1. In this case $g(i_0, i_1) = \log \beta$. From the other hand $u_2(i_1) - u_2(i_0) = -\log(\alpha\beta)$ by Proposition 8.8, which is at least $\log \beta$ because of the inequality (8.4).

Proceed to the second case, where there is no crossing between i_0 and i_1. The smallest part is always the same, therefore

$$g(i_0, i_1) = e(i_0, i_1) \log \alpha + f(i_0, i_1) \log \beta = \log \left(\frac{x(i_1)}{x(i_0)} \right).$$

Denote $v_2(i_1) - v_2(i_0)$ and $u_2(i_1) - u_2(i_0)$ as Δv_2 and Δu_2 correspondingly. Since $T(i_0)$ and $T(i_1)$ are central, we have

$$\begin{aligned} \Delta u_2 &= -\log(x(i_1)y(i_1)) + \log(x(i_0)y(i_0)) \\ &= -\log y(i_1) + \log y(i_0) - \log \left(\frac{x(i_1)}{x(i_0)} \right) \\ &= \Delta v_2 - g(i_0, i_1)\,, \end{aligned}$$

which is equivalent to

$$g(i_0, i_1) = \Delta v_2 - \Delta u_2\,.$$

Using Proposition 8.6.1 and 8.6.2 for moments i_1 and i_0 respectively, we obtain

$$g(i_0, i_1) = \Delta v_2 - \Delta u_2 \le \frac{\Delta u_2 + \log\left(\frac{\beta}{\alpha}\right)}{2} - \Delta u_2 = \frac{-\Delta u_2 + \log\left(\frac{\beta}{\alpha}\right)}{2}\,.$$

Recall that $y(i_0) < 1/(2\beta)$, which implies that $y(i_0 + 1) < \frac{1}{2}$, i.e., $T(i_0 + 1)$ is central. Using Proposition 8.8, we deduce

$$\Delta u_2 = u_2(i_1) - u_2(i_0 + 1) + u_2(i_0 + 1) - u_2(i_0) \ge -\log(\alpha\beta)\,.$$

Using this inequality and the inequality (8.4), we have

$$g(i_0, i_1) \le \frac{-\Delta u_2 + \log\left(\frac{\beta}{\alpha}\right)}{2} \le \frac{\log(\alpha\beta) + \log\left(\frac{\beta}{\alpha}\right)}{2} = \log \beta \le -\log(\alpha\beta) \le \Delta u_2\,.$$

■

Proof of Proposition 8.3. Recall that $u_2(i) \ge u_1(i)$ for any $x(i)$, $y(i)$ and $t(i)$ by Proposition 8.5.5.

If $u_1(i-1) < u_1'$, then $u_1(i) \ge u_1(i-1)$ by Proposition 8.7 and therefore we have $u(i) \ge u_1(i) \ge u_1(i-1) = u(i-1)$. If $T(i-1)$ is central and there is a crossing, then $u_1(i) - u_1(i-1) \ge \log \beta$ by Proposition 8.7, hence, $u(i) - u(i-1) \ge \log \beta$.

If $u_1(i-1) \geq u_1'$, then $u_1(i) \geq u_1'$ by Proposition 8.7 and $u_2(i) \geq u_2(i-1)$ by Proposition 8.8. It means that $u(i) = u_2(i) \geq u_2(i-1) = u(i-1)$. If $T(i-1)$ is central and there is a crossing, then $T(i)$ is also central by Proposition 8.6.5. So, we can use Proposition 8.8, which gives us $u_2(i) - u_2(i-1) \geq -\log(\alpha\beta)$. Since $\alpha\beta^2 \leq 1$(see (8.4)), this expression is not less than $\log\beta$.

■

Proof of Proposition 8.2. Recall that $v_2(i) \geq v_1(i)$ for any $x(i)$, $y(i)$ and $t(i)$ by Proposition 8.5.6.

Suppose $u_1(i-1) < u_1''$. If there is no crossing, then the statement is implied by Proposition 8.9. If there is a crossing, then Proposition 8.9 gives us $v_1(i) > v_1(i-1)$, therefore, $v(i) \geq v_1(i) > v_1(i-1) = v(i-1)$, i.e., $v(i) - v(i-1) > 0$. For central $T(i-1)$ this bound is good enough. It remains to consider the case when $T(i-1)$ is not central. From $u_1(i-1) < u_1''$ we have $v(i)-v(i-1) \geq v_1(i)-v(i-1) = v_1(i)-v_1(i-1)$. Since there is a crossing, it is clear that transmission is correct. Condition $T(i-1)$ is not central implies that $t(i) = \beta t(i-1)$ and $v(i) - v(i-1) \geq v_1(i) - v_1(i-1) = \log\beta$.

Suppose $u_1(i-1) \geq u_1''$. In this case $u_1(i) \geq u_1''$ by Proposition 8.7. If there is no crossing, then proposition follows from Proposition 8.4. If there is a crossing, then $T(i-1)$ is central by Proposition 8.6.6. In this case $v(i)-v(i-1) = v_2(i)-v_2(i-1) \geq -\log\beta$ by Proposition 8.4.

■

9
Further Publications by the Author

We supplement this thesis with a summary of selected additional works the author contributed to throughout his PhD.

9.1 FuLeeca

This section is based on [RMB+23] and the algorithmic specification [RBK+] which his been submitted for the additional call of NIST for digital signatures and is joint work with all coauthors of the mentioned works, forming the FuLeeca team.

FuLeeca is a code-based signature scheme that bases its security on the NP-hardness of finding codewords of small Lee weight if a generator matrix in row reduced echelon form is given. The scheme follows the Hash-and-Sign paradigm with quasi-cyclic codes in the Lee metric. The reason for choosing the Lee-metric is to thwart known attacks used to break similar schemes that are defined over the Hamming metric. Previously standardized schemes by NIST base their security on the hardness of mathematical problems related to structured lattices or the hardness of finding preimages or collisions for outputs of cryptographic hash functions. In its call for alternative signatures NIST explicitly asks designers to provide schemes that base their security on alternative hard mathematical problems. FuLeeca not only fulfills this alternative hardness property but furthermore features small signature and key sizes. The scheme however has been broken in [HW24] using an inherent property of our scheme that permits an attacker to get the signing key by searching for a short vector in a circulant integer lattice. Even though this attack breaks FuLeeca constructing signature schemes that base their security on the hardness of finding codewords of small Lee weight is an interesting direction for further research.

9.2 Signature Codes for the Multiple Access Channel

The results presented in this section are based on [EMP22] and have been published in the proceedings of the 2022 IEEE Information Theory Workshop (ITW). The results have been obtained throughout joint work with Gökberk Erdoğan and Nikita Polyanskii.

In this work q-ary signature codes for the noisy multiple adder channel are considered. Imagine a scenario where a large number of devices can send a signal (codeword) to a single destination. We assume that their signals are added up by interference at the destination. On reception of the signal the destination's task is to detect which devices contributed to the received signal. Signature codes have the property the sum of any subset of its codewords is distinct. Thus, the sum gives sufficient information to determine all codewords contributing to the sum. In order to find out which devices are active each device gets a unique codeword of the signature code. If each active device sends its codeword and the non-active devices do not send anything or equivalently the all-zero codeword, the destination is able to identify the active devices. In this work, we are particularly interested in signature codes that are able to tolerate a fraction of τ erroneous positions in the multiple adder channel's output while still uniquely distinguishing any subset of active devices. We provide existence and converse results for codes of length k and cardinality n if the fraction of erroneous channel outputs is bounded by τ over a q-ary code alphabet. Furthermore, an explicit construction for a signature codes with the ability to correct a fixed number of t errors as well as an explicit construction being able to correct a fraction of $\approx (q - 1/8q)k$ errors is given.

10 Conclusion

In Part I of this thesis it has been shown that information theoretic concepts can be used to analyze public key encryption schemes and Physical Unclonable Functions in the asymptotic as well as the finite blocklength regime. We have characterized the channel resulting from the concatenation of encryption and decryption for Learning With Errors (LWE) and we have computed a lower bound on its capacity based on investigating the marginalization of the channel. We motivated why we assume that this lower bound is reasonably tight and showed that using error correcting codes over non-binary alphabets can lead to a substantial increase in the achievable bitrate of the marginalized channel in the asymptotic setting. Furthermore, we have considered the problem of minimizing the ciphertext size for a fixed bitrate using similar methods. We assumed that the achievable rate for the marginalized channel is also a lower bound on the LWE/MLWE channel for finite blocklength. Our results show that increasing the alphabet size and using ECCs also enables the reduction of ciphertext and key sizes for a fixed plaintext length. A more concise justification or proof for the assumptions made in the finite blocklength regime is an interesting topic for further research.

The second topic covered in Part I has been the derivation of fundamental design limits of PUFs used for tamper protection and key storage. We have shown that active attacks on those PUFs can be prevented using wiretap coding. We analyzed two attack scenarios. The first one featured less sophisticated measurements but kept a larger proportion of the PUF intact, thereby enabling the measurement of more PUF cells. The second attack uses more sophisticated measurement tools and combines the measurement of the first attack scenario with additional measurements. We provide asymptotic results on the existence of wiretap codes to prevent an attacker from retrieving information about the secret key. Furthermore, we used finite length bounds to obtain achievability and converse bounds on the required amount of PUF cells for targetted PUF reliability and security levels. By examining the converse we have shown that existing schemes proposed in the literature cannot achieve the reliability and security levels required for state of the art PUF based tamper protection.

In Part II of this thesis coding with feedback for several channels has been inves-

tigated. In this work adversarial error models have been considered. In particular the encoder's task is to communicate a message error free if the maximal amount of adversarial errors scales linearly with the number of channel uses n, i.e. the number of errors t is at most $t = \tau n$ and $\tau \in [0, 1]$. For all channels we considered the capacity error function which specifies the maximal achievable rate for the given error model as the number of channel uses goes to infinity.

First the Z-channel has been examined. We proposed an encoding strategy that consists of two phases. The first phase partitions the message space and is repeated until the remaining error budget is sufficiently small or the set of possible messages has been adequately reduced such that switching to the second phase guarantees successful decoding. In this second phase the remaining possible messages are enumerated. If the attacker can still inflict errors, i.e. the attacker has some error budget left, the index of the correct message is encoded by the Hamming weight of the remaining output sequence. Otherwise the index is just encoded as a binary number and directly sent over the channel. Throughout both phases the property of the Z-channel that reception of a 1 symbol at the decoder guarantees error free transmission of the symbol is utilized. The upper bound on the capacity error function considers messages, leading the output sequences of relatively large Hamming weight combined with a sphere-packing argument. For a large range of the maximal error fraction τ, upper and lower bound are still far from equal. We conjecture that the lower bound is tight and therefore consider especially the construction of a tighter upper bound an interesting research topic.

Next we examined unidirectional channels. Those are composed of two asymmetric channels, one only allowing positive errors and one only allowing negative errors. At the beginning of the block transmission one of the two channels is selected. Throughout this block the errors obey the properties of the selected channel. However, sender and receiver are not given information about the selection prior to the block transmission. Due to the feedback the sender knows the selected channel once the first error occurs and may adapt his encoding strategy to forward this information also to the decoder. We have shown upper and lower bounds on the capacity error function for those channels. For channels with odd alphabet sizes we obtain the capacity error function for $\tau \in [1/2, 1]$. For input alphabet sizes greater than 3 we obtain lower and upper bounds for the capacity error functions. Furthermore, we have shown that an adaptation of the rubber method is possible for binary alphabets, achieving the same asymptotic rate as possible for the Z-channel. Thereby, we established that communicating to the receiver which channel has been selected implies no loss in rate for the rubber method.

Finally, we examined the insertion-deletion channel with noiseless feedback for binary input and output. In this thesis we have shown how to completely obtain the capacity error function of this channel. In proving this result, we make use of the well known capacity error function of the symmetric channel with binary alphabets. We

obtain an upper bound on the capacity error function by showing that it is possible to construct a code for the symmetric channel from a code for the insertion-deletion channel. Similarly, we have shown that it is possible to construct an encoding strategy for the insertion-deletion channel from an encoding strategy for the symmetric channel. The achievable asymptotic rate obtained by applying the encoding strategy originally analyzed for the symmetric channel to the insertion-deletion channel matches the upper bound. Therefore, the capacity error function has been established.

Related Publications by the Author

[DLM21] Christian Deppe, Vladimir Lebedev, and Georg Maringer. "Bounds for the capacity error function for unidirectional channels with noiseless feedback". In: *Theoretical Computer Science* 856 (2021), pp. 1–13.

[DLMP20] Christian Deppe, Vladimir Lebedev, Georg Maringer, and Nikita Polyanskii. "Coding with noiseless feedback over the Z-channel". In: *International Computing and Combinatorics Conference*. Springer. 2020, pp. 98–109.

[DLMP22] Christian Deppe, Vladimir Lebedev, Georg Maringer, and Nikita Polyanskii. "Coding with Noiseless Feedback over the Z-channel". In: *IEEE Transactions on Information Theory* 68.6 (2022), pp. 3731–3739.

[DML20] Christian Deppe, Georg Maringer, and Vladimir Lebedev. "Bounds for the capacity error function for unidirectional channels with noiseless feedback". In: *2020 IEEE International Symposium on Information Theory (ISIT)*. 2020, pp. 2061–2066. DOI: 10.1109/ISIT44484.2020.9174449.

[EMP22] Gökberk Erdoğan, Georg Maringer, and Nikita Polyanskii. "Signature Codes for a Noisy Adder Multiple Access Channel". In: *2022 IEEE Information Theory Workshop (ITW)*. 2022, pp. 476–481. DOI: 10.1109/ITW54588.2022.9965762.

[MPVW21a] Georg Maringer, Nikita Polyanskii, Ilya Vorobyev, and Lorenz Welter. "Feedback Insertion-Deletion Codes". In: *2020 IEEE Information Theory Workshop (ITW)*. 2021, pp. 1–5. DOI: 10.1109/ITW46852.2021.9457653.

[MPVW21b] Georg Maringer, Nikita Andreevich Polyanskii, Ilya V Vorobyev, and Lorenz Welter. "Feedback insertion-deletion codes". In: *Problems of Information Transmission* 57 (2021), pp. 212–240.

[MPWZ21] Georg Maringer, Sven Puchinger, and Antonia Wachter-Zeh. "Higher rates and information-theoretic analysis for the RLWE channel". In: *2020 IEEE Information Theory Workshop (ITW)*. IEEE. 2021, pp. 1–5.

[MPWZ22] Georg Maringer, Sven Puchinger, and Antonia Wachter-Zeh. "Information-and Coding-Theoretic Analysis of the RLWE/MLWE Channel". In: *IEEE Transactions on Information Forensics and Security* 18 (2022), pp. 549–564.

[MXP+21] Georg Maringer, Marvin Xhemrishi, Sven Puchinger, Kathrin Garb, Hedongliang Liu, Thomas Jerkovits, Ludwig Kürzinger, Matthias Hiller, and Antonia Wachter-Zeh. "Analysis of Communication Channels Related to Physical Unclonable Functions". In: *arXiv preprint arXiv:2112.02198* (2021).

[RBK+] Stefan Ritterhoff, Sebastian Bitzer, Patrick Karl, Georg Maringer, Thomas Schamberger, Jonas Schupp, Georg Sigl, Antonia Wachter-Zeh, and Violetta Weger. "Submission to the NIST Post-Quantum Cryptography Standardization Process Algorithm Specifications and Supporting Documentation". In: *NIST PQC Additional Call for Signatures* (). `https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures`.

[RMB+23] Stefan Ritterhoff, Georg Maringer, Sebastian Bitzer, Violetta Weger, Patrick Karl, Thomas Schamberger, Jonas Schupp, and Antonia Wachter-Zeh. "FuLeeca: A Lee-based signature scheme". In: *Code-Based Cryptography Workshop*. Springer. 2023, pp. 56–83.

Bibliography

[AAB+19] Erdem Alkim, Roberto Avanzi, Joppe Bos, Léo Ducas, Antonio de la Piedra, Thomas Pöppelmann, Peter Schwabe, and Douglas Stebila. *Newhope: Algorithm specification and supporting documentation. submission to the nist post-quantum cryptography standardization project, 2017*. 2019.

[AAKT06] Rudolf Ahlswede, H Aydinian, Levon H Khachatrian, and Ludo MGM Tolhuizen. "On q-ary codes correcting all unidirectional errors of a limited magnitude". In: *arXiv preprint cs/0607132* (2006).

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[AB75] R.C. Agarwal and C.S. Burrus. "Number theoretic transforms to implement fast digital convolution". In: *Proceedings of the IEEE* 63.4 (1975), pp. 550–560. DOI: 10.1109/PROC.1975.9791.

[ABB+22] Nicolas Aragon, Paulo Barreto, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Santosh Ghosh, Shay Gueron, Tim Güneysu, et al. "BIKE: bit flipping key encapsulation". In: *NIST PQC Round 4* (2022).

[ABC+22] Martin R Albrecht, Daniel J Bernstein, Tung Chou, Carlos Cid, Jan Gilcher, Tanja Lange, Varun Maram, Ingo Von Maurich, Rafael Misoczki, Ruben Niederhagen, et al. "Classic McEliece: conservative code-based cryptography". In: *NIST PQC Round 4* (2022).

[ABD+20] Erdem Alkim, Joppe Bos, Léo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, and Douglas Stebila. "FrodoKEM, Learning With Errors Key Encapsulation, Algorithmic Specifications and Supporting Documentation". In: *NIST PQC Round 3* (2020). https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions.

[ABD+21] Roberto Avanzi, Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Kyber: Algorithm Specifications And Supporting Documentation". In: *NIST PQC Round 3* (2021). `https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions`.

[AC93] Rudolf Ahlswede and Imre Csiszár. "Common randomness in information theory and cryptography. I. Secret sharing". In: *IEEE Transactions on Information Theory* 39.4 (1993), pp. 1121–1132.

[AD97] Miklós Ajtai and Cynthia Dwork. "A public-key cryptosystem with worst-case/average-case equivalence". In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*. 1997, pp. 284–293.

[ADL05] Rudolf Ahlswede, Christian Deppe, and Vladimir Lebedev. "Non-binary error correcting codes with noiseless feedback, localized errors, or both". In: *Annals of the European Academy of Sciences*. Vol. 1. 4. 2005, pp. 285–309.

[ADL06] Rudolf Ahlswede, Christian Deppe, and Vladimir Lebedev. "Non-binary error correcting codes with noiseless feedback, localized errors, or both". In: *Information Theory, 2006 IEEE International Symposium on*. 2006, pp. 2486–2487.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. "NewHope without reconciliation." In: *IACR Cryptol. ePrint Arch.* 2016 (2016), p. 1157.

[Ajt96] Miklós Ajtai. "Generating hard instances of lattice problems". In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 99–108.

[Ajt98] Miklós Ajtai. "The shortest vector problem in L2 is NP-hard for randomized reductions". In: *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 1998, pp. 10–19.

[APS15] Martin R Albrecht, Rachel Player, and Sam Scott. "On the concrete hardness of learning with errors". In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203.

[Ari09] Erdal Arikan. "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels". In: *IEEE Transactions on Information Theory* 55.7 (2009), pp. 3051–3073. DOI: 10.1109/TIT.2009.2021379.

[Bas65] Leonid Alexandrovich Bassalygo. "New upper bounds for error correcting codes". In: *Problemy Peredachi Informatsii* 1.4 (1965), pp. 41–44.

[BB11] Matthieu Bloch and Joao Barros. *Physical-layer security: from information theory to security engineering*. Cambridge University Press, 2011.

[BBKN12] Alessandro Barenghi, Luca Breveglieri, Israel Koren, and David Naccache. "Fault Injection Attacks on Cryptographic Devices: Theory, Practice, and Countermeasures". In: *Proceedings of the IEEE* 100.11 (2012), pp. 3056–3076. DOI: 10.1109/JPROC.2012.2188769.

[BCD+16] Joppe Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. "Frodo: Take off the ring! practical, quantum-secure key exchange from LWE". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 1006–1018.

[BDH+10] Ileana Buhan, Jeroen Doumen, Pieter Hartel, Qian Tang, and Raymond Veldhuis. "Embedding renewable cryptographic keys into noisy data". In: *International Journal of Information Security* 9 (2010), pp. 193–208.

[BDK+18] Joppe Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM". In: *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2018, pp. 353–367.

[BDL97] Dan Boneh, Richard A DeMillo, and Richard J Lipton. "On the importance of checking cryptographic protocols for faults". In: *International conference on the theory and applications of cryptographic techniques*. Springer. 1997, pp. 37–51.

[Ber64] Elwyn R Berlekamp. "Block coding with noiseless feedback". PhD thesis. Massachusetts Institute of Technology, 1964.

[Ber68] E. R. Berlekamp. "Block coding for the binary symmetric channel with noiseless, delayless feedback". In: *Error correcting codes*. Ed. by Henry B. Mann. Publications of the Mathematics Research Center, United States Army, University of Wisconsin 21. (Madison, WI, 6–8 May 1968). A Russian translation was published in *Kibern. Sb.* **9** (1972). MR:0234756. Zbl:0176.49404. New York: John Wiley, 1968, pp. 61–88. ISBN: 9780471567158.

[Ber95] Claude Berrou. *Error-correction coding method with at least two systematic convolutional codings in parallel, corresponding iterative decoding method, decoding module and decoder*. US Patent 5,446,747. 1995.

[BG16] Boris Bukh and Venkatesan Guruswami. "An improved bound on the fraction of correctable deletions". In: *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*. 2016, pp. 1893–1901.

[BGH16] Boris Bukh, Venkatesan Guruswami, and Johan Håstad. "An improved bound on the fraction of correctable deletions". In: *IEEE Transactions on Information Theory* 63.1 (2016), pp. 93–103.

[BGML+18] Sauvik Bhattacharya, Oscar Garcia-Morchon, Thijs Laarhoven, Ronald Rietman, Markku-Juhani O Saarinen, Ludo Tolhuizen, and Zhenfei Zhang. "Round5: Compact and Fast Post-Quantum Public-Key Encryption." In: *IACR Cryptol. ePrint Arch.* 2018 (2018), p. 725.

[BGT93] Claude Berrou, Alain Glavieux, and Punya Thitimajshima. "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1". In: *Proceedings of ICC'93-IEEE International Conference on Communications*. Vol. 2. IEEE. 1993, pp. 1064–1070.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) fully homomorphic encryption without bootstrapping". In: *ACM Transactions on Computation Theory (TOCT)* 6.3 (2014), pp. 1–36.

[BH22] Jakub Breier and Xiaolu Hou. "How Practical Are Fault Injection Attacks, Really?" In: *IEEE Access* 10 (2022), pp. 113122–113130. DOI: 10.1109/ACCESS.2022.3217212.

[BHK+19] Daniel J Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. "The SPHINCS+ signature framework". In: *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. 2019, pp. 2129–2146.

[BHK+22] Irina E Bocharova, Henk DL Hollmann, Karan Khathuria, Boris D Kudryashov, and Vitaly Skachek. "Coding with Cyclic PAM and Vector Quantization for the RLWE/MLWE Channel". In: *2022 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2022, pp. 666–671.

[Bla94] Mario Blaum. *Codes for detecting and correcting unidirectional errors*. IEEE Computer Society Press, 1994.

[BMD+] Andrea Basso, Jose Maria Bermudo Mera, Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Michiel Van Beirendonck, and Frederik Vercauteren. "SABER: Mod-LWR based KEM (Round 3 Submission)". In: (). https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[BMT78] E. Berlekamp, R. McEliece, and H. van Tilborg. "On the inherent intractability of certain coding problems (Corresp.)" In: *IEEE Transactions on Information Theory* 24.3 (1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

[Bos13] Martin Bossert. *Kanalcodierung*. Oldenbourg Wissenschaftsverlag Verlag, 2013.

[BPR12] Abhishek Banerjee, Chris Peikert, and Alon Rosen. "Pseudorandom functions and lattices". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2012, pp. 719–737.

[Bra16] Matt Braithwaite. *Experimenting with Post-Quantum Cryptography*. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html. 2016.

[BRC60] Raj Chandra Bose and Dwijendra K Ray-Chaudhuri. "On a class of error correcting binary group codes". In: *Information and control* 3.1 (1960), pp. 68–79.

[Bru77] Richard A. Brualdi. *Introductory Combinatorics*. Elsevier, North-Holland, 1977.

[BS99] Johannes Blömer and Jean-Pierre Seifert. "On the complexity of computing short linearly independent vectors and short bases in a lattice". In: *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. 1999, pp. 711–720.

[BSI24] BSI. *TR-02102-1: "Cryptographic Mechanisms: Recommendations and Key Lengths"*. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html. [Version 2024-1]. 2024.

[CDH+] Chong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hülsing, Joost Rijneveld, John M. Schanck, Tsunekazu Saito, Peter Schwabe, William Whyte, Keita Xagawa, Takashi Yamakawa, and Zhenfei Zhang. "NTRU, Algorithmic Specifications And Supporting Documentation". In: *NIST PQC Round 3* (). https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[Cic13] Ferdinando Cicalese. *Fault-Tolerant Search Algorithms - Reliable Computation with Unreliable Information*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2013. ISBN: 978-3-642-17326-4. DOI: 10.1007/978-3-642-17327-1. URL: https://doi.org/10.1007/978-3-642-17327-1.

[CIW+17] Bin Chen, Tanya Ignatenko, Frans M J. Willems, Roel Maes, Erik van der Sluis, and Georgios Selimis. "A Robust SRAM-PUF Key Generation Scheme Based on Polar Codes". In: *IEEE Global Communications Conference (GLOBECOM)*. 2017.

[CJW06] James Carlson, Arthur Jaffe, and Andrew Wiles. *The Millenium Prize Problems*. American Mathematical Society, 2006.

[CK78] Imre Csiszár and Janos Korner. "Broadcast channels with confidential messages". In: *IEEE transactions on information theory* 24.3 (1978), pp. 339–348.

[CM00] Ferdinando Cicalese and Daniele Mundici. "Optimal coding with one asymmetric error: below the sphere packing bound". In: *International Computing and Combinatorics Conference*. Springer. 2000, pp. 159–169.

[CR20] Mahdi Cheraghchi and João Ribeiro. "An overview of capacity results for synchronization channels". In: *IEEE Transactions on Information Theory* (2020).

[CSBB10] Yuval Cassuto, Moshe Schwartz, Vasken Bohossian, and Jehoshua Bruck. "Codes for asymmetric limited-magnitude errors with application to multilevel flash memories". In: *IEEE Transactions on Information theory* 56.4 (2010), pp. 1582–1595.

[CT99] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 1999.

[DCRVV15] Ruan De Clercq, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. "Efficient software implementation of ring-LWE encryption". In: *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE. 2015, pp. 339–344.

[DFKL+20] Luca De Feo, David Kohel, Antonin Leroux, Christophe Petit, and Benjamin Wesolowski. "SQISign: compact post-quantum signatures from quaternions and isogenies". In: *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*. Springer. 2020, pp. 64–93.

[DGSV15] Jeroen Delvaux, Dawu Gu, Dries Schellekens, and Ingrid Verbauwhede. "Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34.6 (2015), pp. 889–902.

[DGV+16] Jeroen Delvaux, Dawu Gu, Ingrid Verbauwhede, Matthias Hiller, and Mandel Yu. "Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications". In: *Conference on Cryptographic Hardware and Embedded Systems (CHES)*. Ed. by Benedikt Gierlichs and Axel Poschmann. Vol. 9813. LNCS. Springer Berlin / Heidelberg, 2016, pp. 412–431.

[DH76] W. Diffie and M. Hellman. "New directions in cryptography". In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[DKL+00] Jean-Francois Dhem, Francois Koeune, Philippe-Alexandre Leroux, Patrick Mestré, Jean-Jacques Quisquater, and Jean-Louis Willems. "A practical implementation of the timing attack". In: *Smart Card Research and Applications: Third International Conference, CARDIS'98, Louvain-la-Neuve, Belgium, September 14-16, 1998. Proceedings 3*. Springer. 2000, pp. 167–182.

[DKRV18] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. "Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM". In: *International Conference on Cryptology in Africa*. Springer. 2018, pp. 282–305.

[DL19] Christian Deppe and Vladimir Lebedev. "Algorithms for Q-ary Error-Correcting Codes with Partial Feedback and Limited Magnitude". In: *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2019, pp. 2244–2248.

[DR98] Joan Daemen and Vincent Rijmen. "The block cipher Rijndael". In: *International Conference on Smart Card Research and Advanced Applications*. Springer. 1998, pp. 277–284.

[DRS04a] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data". In: *International conference on the theory and applications of cryptographic techniques*. Springer. 2004, pp. 523–540.

[DRS04b] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data". In: *Advances in Cryptology (EUROCRYPT)*. Ed. by Christian Cachin and Jan L. Camenisch. Vol. 3027. LNCS. Springer Berlin / Heidelberg, 2004, pp. 523–540.

[DS04] Ioana Dumitriu and Joel Spencer. "A Halfliar's game". In: *Theoretical computer science* 313.3 (2004), pp. 353–369.

[DVV19] Jan-Pieter D'Anvers, Frederik Vercauteren, and Ingrid Verbauwhede. "The impact of error dependencies on Ring/Mod-LWE/LWR based schemes". In: *International Conference on Post-Quantum Cryptography*. Springer. 2019, pp. 103–115.

[D'y75] Arkadii Georgievich D'yachkov. "Upper bounds on the error probability for discrete memoryless channels with feedback". In: *Problemy Peredachi Informatsii* 11.4 (1975), pp. 13–28.

[EFK+12] Thomas Esbach, Walter Fumy, Olga Kulikovska, Dominik Merli, Dieter Schuster, and Frederic Stumpf. "A new security architecture for smartcards utilizing PUFs". In: *ISSE 2012 Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe 2012 Conference*. Springer. 2012, pp. 180–194.

[FHK+18] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang, et al. "Falcon: Fast-Fourier lattice-based compact signatures over NTRU". In: *Submission to the NIST's post-quantum cryptography standardization process* 36.5 (2018), pp. 1–75.

[Flu16] Scott R Fluhrer. "Cryptanalysis of ring-LWE based key exchange with key share reuse." In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 85.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. "Secure integration of asymmetric and symmetric encryption schemes". In: *Annual International Cryptology Conference*. Springer. 1999, pp. 537–554.

[FPS18] Tim Fritzmann, Thomas Pöppelmann, and Johanna Sepulveda. "Analysis of error-correcting codes for lattice-based key exchange". In: *International Conference on Selected Areas in Cryptography*. Springer. 2018, pp. 369–390.

[FSVVM+18] Josep Font-Segura, Gonzalo Vazquez-Vilar, Alfonso Martinez, Albert Guillén i Fàbregas, and Alejandro Lancho. "Saddlepoint approximations of lower and upper bounds to the error probability in channel coding". In: *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2018, pp. 1–6.

[FWHP23] Christoph Frisch, Florian Wilde, Thomas Holzner, and Michael Pehl. "A Practical Approach to Estimate the Min-Entropy in PUFs". In: *Journal of Hardware and Systems Security* (2023), pp. 1–9.

[Gal62] R. Gallager. "Low-density parity-check codes". In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28. DOI: 10.1109/TIT.1962.1057683.

[GC99] Constantino Goutis and George Casella. "Explaining the saddlepoint approximation". In: *The American Statistician* 53.3 (1999), pp. 216–224.

[GFS+12] Norman Göttert, Thomas Feller, Michael Schneider, Johannes Buchmann, and Sorin Huss. "On the design of hardware building blocks for modern lattice-based encryption schemes". In: *Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings 14*. Springer. 2012, pp. 512–529.

[GGV17] Michael Geis, Karen Gettings, and Michael Vai. "Optical physical unclonable function". In: *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2017, pp. 1248–1251. DOI: 10.1109/MWSCAS.2017.8053156.

[Gil52] Edgar N Gilbert. "A comparison of signalling alphabets". In: *The Bell system technical journal* 31.3 (1952), pp. 504–522.

[GKST07] Jorge Guajardo, Sandeep S Kumar, Geert Jan Schrijen, and Pim Tuyls. "FPGA Intrinsic PUFs and Their Use for IP Protection". In: *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Ed. by Pascal Paillier and Ingrid Verbauwhede. Vol. 4727. LNCS. Springer Berlin / Heidelberg, 2007, pp. 63–80.

[GM02] Shafi Goldwasser and Daniele Micciancio. *Complexity of lattice problems: a cryptographic perspective*. Springer, 2002.

[GM84] Shafi Goldwasser and Silvio Micali. "Probabilistic encryption". In: *Journal of Computer and System Sciences* 28 (1984), pp. 270–299. DOI: https://doi.org/10.1016/0022-0000(84)90070-9.

[GMSS99] Oded Goldreich, Daniele Micciancio, Shmuel Safra, and J-P Seifert. "Approximating shortest lattice vectors is not harder than approximating closest lattice vectors". In: *Information Processing Letters* 71.2 (1999), pp. 55–61.

[GN07] Nicolas Gama and Phong Q Nguyen. "New chosen-ciphertext attacks on NTRU". In: *International Workshop on Public Key Cryptography*. Springer. 2007, pp. 89–106.

[GOFK21] Kathrin Garb, Johannes Obermaier, Elischa Ferres, and Martin König. "FORTRESS: FORtified Tamper-Resistant Envelope with Embedded Security Sensor". In: *International Conference on Privacy, Security and Trust (PST)*. 2021, pp. 1–12.

[Gop70] Valerii Denisovich Goppa. “A new class of linear correcting codes”. In: *Problemy Peredachi Informatsii* 6.3 (1970), pp. 24–30.

[Gro96] Lov K Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 1996, pp. 212–219.

[GSHO21] Kathrin Garb, Marc Schink, Matthias Hiller, and Johannes Obermaier. “Attacks and Countermeasures for Capacitive PUF-Based Security Enclosures”. In: *IEEE Physical Assurance and Inspection of Electronics (PAINE)*. 2021, pp. 1–8.

[GSVL16] Joep de Groot, Boris Skoric, Niels de Vreede, and Jean-Paul Linnartz. “Quantization in zero leakage helper data schemes”. In: *EURASIP Journal on Advances in Signal Processing* 2016.1 (2016), p. 54.

[GXKF22a] Kathrin Garb, Marvin Xhemrishi, Ludwig Kürzinger, and Christoph Frisch. “The Wiretap Channel for Capacitive PUF-Based Security Enclosures”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* (2022), pp. 165–191.

[GXKF22b] Kathrin Garb, Marvin Xhemrishi, Ludwig Kürzinger, and Christoph Frisch. “The Wiretap Channel for Capacitive PUF-Based Security Enclosures”. In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2022.3 (2022), pp. 165–191.

[Ham50] R. W. Hamming. “Error detecting and error correcting codes”. In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x.

[HBNS13] Clemens Helfmeier, Christian Boit, Dmitry Nedospasov, and Jean-Pierre Seifert. “Cloning physically unclonable functions”. In: *2013 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*. IEEE. 2013, pp. 1–6.

[HGNP+03] Nick Howgrave-Graham, Phong Q Nguyen, David Pointcheval, John Proos, Joseph H Silverman, Ari Singer, and William Whyte. “The impact of decryption failures on the security of NTRU encryption”. In: *Annual International Cryptology Conference*. Springer. 2003, pp. 226–246.

[HHK17] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. “A modular analysis of the Fujisaki-Okamoto transformation”. In: *Theory of Cryptography Conference*. Springer. 2017, pp. 341–371.

[HKS20] Matthias Hiller, Ludwig Kürzinger, and Georg Sigl. “Review of error correction for PUFs and evaluation on state-of-the-art FPGAs”. In: *Journal of Cryptographic Engineering* (2020).

[HO17] Matthias Hiller and Aysun Gurur Önalan. "Hiding Secrecy Leakage in Leaky Helper Data". In: *Conference on Cryptographic Hardware and Embedded Systems*. Ed. by Wieland Fischer and Naofumi Homma. Vol. 10529. LNCS. Springer Berlin / Heidelberg, 2017, pp. 601–619.

[Hor63] Michael Horstein. "Sequential transmission using noiseless feedback". In: *IEEE Transactions on Information Theory* 9.3 (1963), pp. 136–143.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A ring-based public key cryptosystem". In: *Algorithmic Number Theory*. Springer Berlin Heidelberg, 1998, pp. 267–288.

[HPSS14] Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*. Vol. 2. Springer, 2014.

[HTW16] Masahito Hayashi, Himanshu Tyagi, and Shun Watanabe. "Secret Key Agreement: General Capacity and Second-Order Asymptotics". In: *IEEE Transactions on Information Theory* 62.7 (2016), pp. 3796–3810. DOI: 10.1109/TIT.2016.2567440.

[HW24] Felicitas Hörmann and Wessel van Woerden. "FuLeakage: Breaking FuLeeca by learning attacks". In: *Annual International Cryptology Conference*. Springer. 2024, pp. 253–286.

[HYKD14] Charles Herder, Mandel Yu, Farinaz Koushanfar, and Srinivas Devadas. "Physical Unclonable Functions and Applications: A Tutorial". In: *Proceedings of the IEEE* 102.8 (2014), pp. 1126–1141.

[IHKS16] Vincent Immler, Maxim Hennig, Ludwig Kürzinger, and Georg Sigl. "Practical Aspects of Quantization and Tamper-Sensitivity for Physically Obfuscated Keys". In: *Workshop on Cryptography and Security in Computing Systems (CS2)*. ACM, 2016, pp. 13–18.

[IHL+18] Vincent Immler, Matthias Hiller, Qinzhi Liu, Andreas Lenz, and Antonia Wachter-Zeh. "Variable-Length Bit Mapping and Error Correcting Codes for Higher-Order Alphabet PUFs". In: *Journal of Hardware and Systems Security (HASS)* 2.4 (2018).

[Imm19] Vincent Immler. "Higher-Order Alphabet Physical Unclonable Functions". Dissertation. 2019.

[IOK+18] Vincent Immler, Johannes Obermaier, Martin König, Matthias Hiller, and Georg Sigl. "B-TREPID: Batteryless Tamper-Resistant Envelope with a PUF and Integrity Detection". In: *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*. 2018, pp. 49–56.

[ION+19] Vincent Immler, Johannes Obermaier, Kuan Kuan Ng, Fei Xiang Ke, Jin Ju Lee, Yak Peng Lim, Wei Koon Oh, Keng Hoong Wee, and Georg Sigl. "Secure Physical Enclosures from Covers with Tamper-Resistance". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019.1 (2019).

[IU19] Vincent Immler and Karthik Uppund. "New Insights to Key Derivation for Tamper-Evident Physical Unclonable Functions". In: *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019.3 (2019), pp. 30–65.

[JJ00] Éliane Jaulmes and Antoine Joux. "A chosen-ciphertext attack against NTRU". In: *Annual international cryptology conference*. Springer. 2000, pp. 20–35.

[Kah96] David Kahn. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. Simon and Schuster, 1996.

[KJJ99] Paul Kocher, Joshua Jaffe, and Benjamin Jun. "Differential power analysis". In: *Advances in Cryptology — CRYPTO' 99*. Springer, 1999.

[KKM+17] Shrinivas Kudekar, Santhosh Kumar, Marco Mondelli, Henry D Pfister, Eren Sasouglu, and Rüdiger L Urbanke. "Reed–Muller codes achieve capacity on erasure channels". In: *IEEE Transactions on information theory* 63.7 (2017), pp. 4298–4316.

[KL21] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Vol. 3. Chapman and hall/CRC, 2021.

[Koc96] Paul C Kocher. "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems". In: *Advances in Cryptology—CRYPTO'96: 16th Annual International Cryptology Conference Santa Barbara, California, USA August 18–22, 1996 Proceedings 16*. Springer. 1996, pp. 104–113.

[Kra+08] Gerhard Kramer et al. "Topics in multi-user information theory". In: *Foundations and Trends® in Communications and Information Theory* 4.4–5 (2008), pp. 265–444.

[Leb16] Vladimir S Lebedev. "Coding with noiseless feedback". In: *Problems of Information Transmission* 52 (2016), pp. 103–113.

[Lee58] C Lee. "Some properties of nonbinary error-correcting codes". In: *IRE Transactions on Information Theory* 4.2 (1958), pp. 77–82.

[Lev66] Vladimir I Levenshtein. "Binary codes capable of correcting deletions, insertions, and reversals". In: *Soviet physics doklady*. Vol. 10. 8. 1966, pp. 707–710.

[LKN+19] Eunsang Lee, Young-Sik Kim, Jong-Seon No, Minki Song, and Dong-Joon Shin. "Modification of Frodokem Using Gray and Error-Correcting Codes". In: *IEEE Access* 7 (2019), pp. 179564–179574.

[LKP09] Yingbin Liang, Gerhard Kramer, and H Vincent Poor. "Compound wiretap channels". In: *EURASIP Journal on Wireless Communications and Networking* 2009 (2009), pp. 1–12.

[LLJ+17a] Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, and Zhenfei Zhang. *Supporting documentation: LAC*. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions`. 2017.

[LLJ+17b] Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, and Zhenfei Zhang. *Supporting documentation: LAC*. `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions`. 2017.

[LP11] Richard Lindner and Chris Peikert. "Better key sizes (and attacks) for LWE-based encryption". In: *Cryptographers' Track at the RSA Conference*. Springer. 2011, pp. 319–339.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On ideal lattices and learning with errors over rings". In: *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*. Springer. 2010, pp. 1–23.

[LPR13] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On ideal lattices and learning with errors over rings". In: *Journal of the ACM (JACM)* 60.6 (2013), pp. 1–35.

[LSSR+15] Zhe Liu, Hwajeong Seo, Sujoy Sinha Roy, Johann Großschädl, Howon Kim, and Ingrid Verbauwhede. "Efficient Ring-LWE encryption on 8-bit AVR processors". In: *Cryptographic Hardware and Embedded Systems–CHES 2015: 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings 17*. Springer. 2015, pp. 663–682.

[LTBS16] Heiko Lohrke, Shahin Tajik, Christian Boit, and Jean-Pierre Seifert. "No place to hide: Contactless probing of secret data on FPGAs". In: *Cryptographic Hardware and Embedded Systems–CHES 2016: 18th International Conference, Santa Barbara, CA, USA, August 17-19, 2016, Proceedings 18*. Springer. 2016, pp. 147–167.

[MAB+18] Carlos Aguilar Melchor, Nicolas Aragon, Slim Bettaieb, Loıc Bidoux, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, Edoardo Persichetti, Gilles Zémor, and IC Bourges. "Hamming quasi-cyclic (HQC)". In: *NIST PQC Round* 2 (2018), pp. 4–13.

[Mae12] Roel Maes. "Physically Unclonable Functions: Constructions, Properties and Applications". Dissertation. 2012.

[Mas98] James L Massey. "Applied digital information theory". In: *lecture notes, ETH Zurich.[Online]. Available: http://www. isiweb. ee. ethz. ch/archive/massey scr/adit1. pdf* (1998).

[Mau93] U.M. Maurer. "Secret key agreement by public discussion from common information". In: *IEEE Transactions on Information Theory* 39.3 (1993), pp. 733–742. DOI: 10.1109/18.256484.

[McE78] Robert J McEliece. "A public-key cryptosystem based on algebraic". In: *Deep Space Network Progress Report* 42-44 (1978), pp. 114–116.

[Mer78] Ralph C Merkle. "Secure communications over insecure channels". In: *Communications of the ACM* 21.4 (1978), pp. 294–299.

[MF11] Alfonso Martinez and Albert Guillén i Fabregas. "Saddlepoint approximation of random-coding bounds". In: *2011 Information Theory and Applications Workshop*. IEEE. 2011, pp. 1–6.

[MFS20] Georg Maringer, Tim Fritzmann, and Johanna Sepúlveda. "The influence of LWE/RLWE parameters on the stochastic dependence of decryption failures". In: *International Conference on Information and Communications Security*. Springer. 2020, pp. 331–349.

[MGS13] Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. "A systematic method to evaluate and compare the performance of physical unclonable functions". In: *Embedded systems design with FPGAs* (2013), pp. 245–267.

[MN97] David JC MacKay and Radford M Neal. "Near Shannon limit performance of low density parity check codes". In: *Electronics letters* 33.6 (1997), pp. 457–458.

[Moo20] Dustin Moody. *The Future Is Now: Spreading the Word About Post-Quantum Cryptography*. 2020. URL: https://www.nist.gov/blogs/taking-measure/future-now-spreading-word-about-post-quantum-cryptography.

[MOP08] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks: Revealing the secrets of smart cards*. Vol. 31. Springer Science & Business Media, 2008.

[MS77a] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. North-Holland, 1977.

[MS77b] Florence Jessie MacWilliams and Neil James Alexander Sloane. *The theory of error-correcting codes*. Vol. 16. Elsevier, 1977.

[Mul54] D. E. Muller. "Application of Boolean algebra to switching circuit design and to error detection". In: *Transactions of the I.R.E. Professional Group on Electronic Computers* EC-3.3 (1954), pp. 6–12. DOI: 10.1109/IREPGELC.1954.6499441.

[Nat] National Institute of Standards and Technology (NIST). *Post-Quantum Cryptography Standardization*. URL: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography.

[OI18] Johannes Obermaier and Vincent Immler. "The Past, Present, and Future of Physical Security Enclosures: From Battery-Backed Monitoring to PUF-Based Inherent Security and Beyond". In: *Journal of Hardware and Systems Security (HASS)* (2018).

[OIHS18] Johannes Obermaier, Vincent Immler, Matthias Hiller, and Georg Sigl. "A Measurement System for Capacitive PUF-based Security Enclosures". In: *ACM/IEEE Design Automation Conference (DAC)*. 2018.

[Pet83] Fabien Petitcolas. "La cryptographie militaire". In: *J. des Sci. Militaires* 9 (1883), pp. 161–191.

[PG14] Thomas Pöppelmann and Tim Güneysu. "Towards practical lattice-based public-key encryption on reconfigurable hardware". In: *Selected Areas in Cryptography–SAC 2013: 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers 20*. Springer. 2014, pp. 68–85.

[Plo60] Morris Plotkin. "Binary codes with specified minimum distance". In: *IRE Transactions on Information Theory* 6.4 (1960), pp. 445–450.

[PPV10] Yury Polyanskiy, H Vincent Poor, and Sergio Verdú. "Channel coding rate in the finite blocklength regime". In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2307–2359.

[PRTG02] Ravikanth Pappu, Ben Recht, Jason Taylor, and Neil Gershenfeld. "Physical one-way functions". In: *Science* 297.5589 (2002), pp. 2026–2030.

[Ree54] I. Reed. "A class of multiple-error-correcting codes and the decoding scheme". In: *Transactions of the IRE Professional Group on Information Theory* 4.4 (1954), pp. 38–49. DOI: 10.1109/TIT.1954.1057465.

[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. Baltimore, MD, USA: Association for Computing Machinery, 2005, 84–93. ISBN: 1581139608. DOI: 10.1145/1060590.1060603. URL: https://doi.org/10.1145/1060590.1060603.

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *Journal of the ACM (JACM)* 56.6 (2009), pp. 1–40.

[Ren61] Alfred Renyi. "On a problem of information theory". In: *MTA Mat. Kut. Int. Kozl.* 6.B (1961), pp. 505–516.

[RFB+23] Carl Riehm, Christoph Frisch, Florin Burcea, Matthias Hiller, Michael Pehl, and Ralf Brederlow. "Structured Design and Evaluation of a Resistor-Based PUF Robust Against PVT-Variations". In: *International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*. IEEE. 2023, pp. 93–98.

[RMK+80] Ronald L. Rivest, Albert R. Meyer, Daniel J. Kleitman, Karl Winklmann, and Joel Spencer. "Coping with errors in binary search procedures". In: *Journal of Computer and System Sciences* 20.3 (1980), pp. 396–404.

[Rot06] Ron Roth. *Introduction to Coding Theory*. Cambridge University Press, 2006. DOI: 10.1017/CBO9780511808968.

[RR01] Josyula R Rao and Pankaj Rohatgi. "Empowering side-channel attacks". In: *Cryptology EPrint Archive* (2001).

[RS60] Irving S Reed and Gustave Solomon. "Polynomial codes over certain finite fields". In: *Journal of the society for industrial and applied mathematics* 8.2 (1960), pp. 300–304.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[RTA18] Aswin Rajagopalan, Andrew Thangaraj, and Shweta Agrawal. "Wiretap polar codes in encryption schemes based on learning with errors problem". In: *2018 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2018, pp. 1146–1150.

[Saa17] Markku-Juhani O. Saarinen. *Supporting documentation: HILA5*. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions. 2017.

[SAS17] Taras Stanko, Fitria Nur Andini, and Boris Skoric. "Optimized quantization in zero leakage helper data systems". In: *IEEE Transactions on Information Forensics and Security* 12.8 (2017), pp. 1957–1966.

[Sch00] Werner Schindler. "A timing attack against RSA with the chinese remainder theorem". In: *Cryptographic Hardware and Embedded Systems—CHES 2000: Second International Workshop Worcester, MA, USA, August 17–18, 2000 Proceedings 2*. Springer. 2000, pp. 109–124.

[Sch20] John M Schanck. "An upper bound on the decryption failure rate of static-key NewHope." In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 326.

[Sch71] J Schalkwijk. "A class of simple and optimal strategies for block coding on the binary symmetric channel with noiseless feedback". In: *IEEE Transactions on Information Theory* 17.3 (1971), pp. 283–287.

[Sha48] Claude Elwood Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[Sha49] Claude E Shannon. "Communication theory of secrecy systems". In: *The Bell system technical journal* 28.4 (1949), pp. 656–715.

[Sha56] Claude Elwood Shannon. "The zero error capacity of a noisy channel". In: *IRE Transactions on Information Theory* 2.3 (1956), pp. 8–19.

[Sho94] Peter W Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[Sho99] Peter W Shor. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer". In: *SIAM review* 41.2 (1999), pp. 303–332.

[SI23] SOG-IS. *Joint Interpretation Library – Application of Attack Potential to Hardware Devices with Security Boxes*. `https://www.sogis.eu/documents/cc/domains/hardware_devices/JIL-Application-of-Attack-Potential-to-Hardware-Devices-with-Security-Boxes-v3.1.pdf`. [Version 3.1]. 2023.

[SMKT06] Boris Skoric, S. Maubach, Tom A. M. Kevenaar, and Pim Tuyls. "Information-theoretic analysis of capacitive physical unclonable functions". In: *Journal of Applied Physics* 100.2 (2006), pp. 024902–024902–11.

[STA09] Eren Sasouglu, Emre Telatar, and Erdal Arikan. "Polarization for arbitrary discrete memoryless channels". In: *2009 IEEE Information Theory Workshop*. IEEE. 2009, pp. 144–148.

[Sto02] Norbert Stolte. "Rekursive Codes mit der Plotkin-Konstruktion und ihre Decodierung." PhD thesis. Darmstadt University of Technology, Germany, 2002.

[SY03] Joel Spencer and Catherine H Yan. "The halflie problem". In: *Journal of Combinatorial Theory, Series A* 103.1 (2003), pp. 69–89.

[SZ99] Leonard J Schulman and David Zuckerman. "Asymptotically good codes correcting insertions, deletions, and transpositions". In: *IEEE Transactions on information theory* 45.7 (1999), pp. 2552–2557.

[TABB08] Luca G Tallini, Sulaiman Al-Bassam, and Bella Bose. "Feedback codes achieving the capacity of the Z-channel". In: *IEEE Transactions on Information Theory* 54.3 (2008), pp. 1357–1362.

[TSS+06a] Pim Tuyls, Geert-Jan Schrijen, Boris Skoric, Jan van Geloven, Nynke Verhaegh, and Rob Wolters. "Read-Proof Hardware from Protective Coatings". In: *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*. Ed. by Louis Goubin and Mitsuru Matsui. Vol. 4249. LNCS. Springer Berlin Heidelberg, 2006, pp. 369–383.

[TSS+06b] Pim Tuyls, Geert-Jan Schrijen, Boris Skoric, Jan Van Geloven, Nynke Verhaegh, and Rob Wolters. "Read-proof hardware from protective coatings". In: *Cryptographic Hardware and Embedded Systems-CHES 2006: 8th International Workshop, Yokohama, Japan, October 10-13, 2006. Proceedings 8*. Springer. 2006, pp. 369–383.

[TV15] Ido Tal and Alexander Vardy. "List decoding of polar codes". In: *IEEE transactions on information theory* 61.5 (2015), pp. 2213–2226.

[Ula91] Stanislaw M Ulam. *Adventures of a Mathematician*. Univ of California Press, 1991.

[Var57] Rom Rubenovich Varshamov. "Estimate of the number of signals in error correcting codes". In: *Docklady Akad. Nauk, SSSR* 117 (1957), pp. 739–741.

[Var73] R. Varshamov. "A class of codes for asymmetric channels and a problem from the additive theory of numbers". In: *IEEE Transactions on Information Theory* 19.1 (1973), pp. 92–95. DOI: 10.1109/TIT.1973.1054954.

[VNK+15] Michael Vai, Ben Nahill, Josh Kramer, Michael Geis, Dan Utin, David Whelihan, and Roger Khazan. "Secure architecture for embedded systems". In: *2015 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE. 2015, pp. 1–5.

[VT65] RR Varshamov and GM Tenenholtz. "A code for correcting a single asymmetric error". In: *Automatica i Telemekhanika* 26.2 (1965), pp. 288–292.

[VWN+16] Michael Vai, David J Whelihan, Benjamin R Nahill, Daniil M Utin, Sean R O'Melia, and Roger I Khazan. "Secure embedded systems". In: *Lincoln Laboratory Journal* 22.1 (2016), pp. 110–122.

[Web89] Jacobus Hendricus Weber. "Bounds and constructions for binary block codes correcting asymmetric or unidirectional errors". In: (1989).

[Wei00] Steve H Weingart. "Physical security devices for computer subsystems: A survey of attacks and defenses". In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2000, pp. 302–317.

[WGP18] Florian Wilde, Berndt M Gammel, and Michael Pehl. "Spatial correlation analysis on physical unclonable functions". In: *IEEE Transactions on Information Forensics and Security* 13.6 (2018), pp. 1468–1480.

[Wyn75] Aaron D Wyner. "The wire-tap channel". In: *Bell system technical journal* 54.8 (1975), pp. 1355–1387.

[WZPLW20] Antonia Wachter-Zeh, Nikita Polianskii, Hedongliang Liu, and Lorenz Welter. *Lecture Notes for Channel Coding*. 2020, pp. 1–129.

[YSP16] Wei Yang, Rafael F Schaefer, and H Vincent Poor. "Finite-blocklength bounds for wiretap channels". In: *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2016, pp. 3087–3091.

[YSP19] Wei Yang, Rafael F Schaefer, and H Vincent Poor. "Wiretap channels: Nonasymptotic fundamental limits". In: *IEEE Transactions on Information Theory* 65.7 (2019), pp. 4069–4093.

[ZHS18] Christian Zenger, David Holin, and Lars Steinschulte. "Enclosure-PUF, Tamper Proofing Commodity Hardware and other Applications". In: *35th Chaos Communication Congress (35c3)*. 2018.

[Zig76] Kamil'Shamil'evich Zigangirov. "On the number of correctable errors for transmission over a binary symmetrical channel with feedback". In: *Problemy Peredachi Informatsii* 12.2 (1976), pp. 3–19.