

# Algebraic Techniques for Low Communication Secure Protocols

PROEFSCHRIFT

ter verkrijging van  
de graad van Doctor aan de Universiteit Leiden,  
op gezag van Rector Magnificus prof. mr. P.F. van der Heijden,  
volgens besluit van het College voor Promoties  
te verdedigen op woensdag 11 maart 2009  
klokke 15.00 uur

door

Robbert de Haan

geboren te Amsterdam  
in 1980.

**Samenstelling van de promotiecommissie:**

**Promotor:** Prof. dr. R. Cramer (CWI & Universiteit Leiden)

**Referent:** Prof. dr. R. Ostrovsky (University of California, Los Angeles)

**Overige leden:**

Prof. dr. I. Damgård (Aarhus University)

Prof. dr. H.W. Lenstra (Universiteit Leiden)

Dr. B. Schoenmakers (Technische Universiteit Eindhoven)

Prof. dr. P. Stevenhagen (Universiteit Leiden)

The research in this thesis has been carried out at the national research center for mathematics and computer science in the Netherlands (CWI) and has been partially funded by the Dutch BSIK/BRICKS project PDC1.

Algebraic Techniques for  
Low Communication Secure Protocols

Haan, Robbert de  
Algebraic Techniques for Low Communication Secure Protocols  
Printed by Grafinoord b.v., Krommenie  
AMS 2000 Subj. class. code: 94A60, 11T71  
NUR: 918  
ISBN-10: 90-6196-549-7  
ISBN-13: 978-90-6196-549-7



Centrum Wiskunde & Informatica



Universiteit Leiden

Copyright © R. de Haan, Amsterdam 2009. All rights reserved.

Cover: Secure Message Transmission & Secure Multi-Party Computation.  
Design by Tobias Baanders, CWI.

# Contents

|           |   |           |
|-----------|---|-----------|
| <b>I</b>  | <b>Introduction</b>                                     | <b>1</b>  |
| <b>1</b>  | <b>History and Overview</b>                             | <b>3</b>  |
| 1.1       | Secure Communication . . . . .                          | 3         |
| 1.2       | Secure Computation . . . . .                            | 7         |
| 1.3       | Contributions . . . . .                                 | 14        |
| 1.3.1     | Perfectly Secure Message Transmission . . . . .         | 14        |
| 1.3.2     | Perfectly Secure Multi-Party Computation . . . . .      | 15        |
| <b>2</b>  | <b>Preliminaries</b>                                    | <b>17</b> |
| 2.1       | Basic Coding Theory . . . . .                           | 17        |
| 2.2       | Classical Bounds from Coding Theory . . . . .           | 20        |
| 2.3       | Algebraic Geometry Preliminaries . . . . .              | 23        |
| 2.3.1     | Projective spaces, topologies and varieties . . . . .   | 23        |
| 2.3.2     | Rational functions, valuations and divisors . . . . .   | 24        |
| 2.3.3     | Rational differential forms . . . . .                   | 26        |
| 2.3.4     | $\mathbb{F}_q$ -rationality . . . . .                   | 28        |
| 2.3.5     | Algebraic-Geometric Codes . . . . .                     | 28        |
| 2.4       | Algebraic-Geometric Bounds from Coding Theory . . . . . | 29        |
| 2.5       | Secret Sharing . . . . .                                | 30        |
| 2.5.1     | Definition . . . . .                                    | 31        |
| 2.5.2     | Linear Secret Sharing . . . . .                         | 35        |
| 2.5.3     | Multiplicative Linear Secret Sharing Schemes . . . . .  | 37        |
| 2.5.4     | Threshold Secret Sharing . . . . .                      | 39        |
| 2.5.5     | Limitations of Treshold Schemes . . . . .               | 40        |
| <b>II</b> | <b>Perfectly Secure Message Transmission</b>            | <b>45</b> |
| <b>3</b>  | <b>Background</b>                                       | <b>47</b> |

|          |  |           |
|----------|--|-----------|
| 3.1      | Introduction . . . . .   | 47        |
| 3.2      | Model . . . . .  | 49        |
| 3.3      | Definitions . . . . .  | 50        |
| 3.4      | Historical Overview . . . . .                                  | 51        |
| <b>4</b> | <b>Single-Phase PSMT</b>                                       | <b>53</b> |
| 4.1      | Necessity of $n \geq 3t + 1$ . . . . .                         | 54        |
| 4.2      | Sufficiency of $n = 3t + 1$ . . . . .                          | 54        |
| 4.3      | Communication Lower Bound for $n \geq 3t + 1$ . . . . .        | 55        |
| <b>5</b> | <b>Two-Phase PSMT</b>  | <b>57</b> |
| 5.1      | Sayed and Abu-Amara's Protocol . . . . .                       | 58        |
| 5.1.1    | Protocol $\Pi_i$ . . . . .                                     | 58        |
| 5.1.2    | Information Reconciliation . . . . .                           | 59        |
| 5.1.3    | Privacy Amplification . . . . .                                | 60        |
| 5.1.4    | The Protocol . . . . .   | 60        |
| 5.2      | A Lower Bound for Two-Phase PSMT . . . . .                     | 60        |
| 5.3      | Improved Reliable Transmission . . . . .                       | 61        |
| 5.4      | Communication-Optimal PSMT for $n = 2t + 1$ . . . . .          | 62        |
| 5.4.1    | Sketch of the Techniques Used . . . . .                        | 62        |
| 5.4.2    | Details of Protocol $\hat{\Pi}_i$ . . . . .                    | 63        |
| 5.4.3    | The Protocol . . . . .   | 66        |
| 5.4.4    | Proofs . . . . .   | 66        |
| 5.4.5    | Complexity Analysis . . . . .                                  | 68        |
| 5.5      | Computationally Efficient PSMT with Optimal Overhead . . . . . | 69        |
| 5.5.1    | Round One . . . . .  | 69        |
| 5.5.2    | Round Two . . . . .  | 69        |
| 5.5.3    | Checking for Dependence Modulo $C$ . . . . .                   | 70        |
| 5.6      | Extra Phases Do Not Improve Efficiency . . . . .               | 71        |
| 5.7      | Three-Phase PSMT . . . . .                                     | 72        |
| 5.7.1    | Basic Protocol . . . . .                                       | 72        |
| 5.7.2    | Expected One-Phase PSMT for $n \geq 2t + 1$ . . . . .          | 73        |
| 5.7.3    | Protocol $\Pi'_i$ . . . . .                                    | 73        |
| 5.8      | Two-Phase PSMT for Non-Tight Parameters . . . . .              | 74        |
| 5.8.1    | Sketch of the Protocol . . . . .                               | 75        |
| 5.8.2    | The Two-Phase Message Transmission Setup . . . . .             | 75        |
| 5.8.3    | Parameter Estimation for $\nu$ . . . . .                       | 76        |

|            |  |            |
|------------|--|------------|
| <b>III</b> | <b>Ramp Sharing</b>  | <b>79</b>  |
| <b>6</b>   | <b>Ramp Schemes</b>  | <b>81</b>  |
| 6.1        | Definition . . . . .   | 83         |
| 6.2        | (Strongly) Multiplicative Ramp Schemes . . . . .                   | 84         |
| 6.3        | Conditions for (Strong) Multiplicativity . . . . .                 | 86         |
| <b>7</b>   | <b>Ramp Sharing Based on Random Codes</b>                          | <b>87</b>  |
| 7.1        | Massey’s Secret Sharing From Codes . . . . .                       | 88         |
| 7.2        | Extensions of Massey’s Idea . . . . .                              | 88         |
| 7.3        | Existence and Bounds . . . . .                                     | 90         |
| 7.3.1      | General lower bounds on $T$ . . . . .                              | 91         |
| 7.3.2      | Bounds from (Weakly) Self-Dual Codes . . . . .                     | 94         |
| 7.4        | High Information Rate Ramp Schemes . . . . .                       | 96         |
| 7.4.1      | A High Information Rate Ramp Scheme . . . . .                      | 96         |
| 7.4.2      | A More Fruitful Approach . . . . .                                 | 96         |
| 7.4.3      | High Information Rate Ramp Schemes: Existence and Bounds . . . . . | 98         |
| <b>8</b>   | <b>Ramp Sharing Based on Algebraic Geometry</b>                    | <b>101</b> |
| 8.1        | Classical Ramp Schemes . . . . .                                   | 101        |
| 8.1.1      | Parallel Multiplication . . . . .                                  | 102        |
| 8.1.2      | Extension Field Multiplication . . . . .                           | 103        |
| 8.2        | Algebraic-Geometric Ramp Schemes . . . . .                         | 106        |
| 8.2.1      | Interpolation in Riemann-Roch spaces . . . . .                     | 106        |
| 8.2.2      | Parallel Multiplication . . . . .                                  | 107        |
| 8.2.3      | Extension Field Multiplication . . . . .                           | 109        |
| 8.2.4      | Multiplication Properties . . . . .                                | 110        |
| 8.2.5      | Achieving Constant Field Size . . . . .                            | 112        |
| <b>IV</b>  | <b>Protocols</b>   | <b>115</b> |
| <b>9</b>   | <b>Basics of Secure Multi-Party Computation</b>                    | <b>117</b> |
| 9.1        | Protocol Structure . . . . .                                       | 118        |
| 9.2        | Addition and Multiplication in the Passive Case . . . . .          | 118        |
| 9.3        | Low Communication MPC with Ramp Schemes . . . . .                  | 120        |
| 9.4        | Formal Security Definitions for MPC . . . . .                      | 121        |
| <b>10</b>  | <b>Active-Adversary Multi-Party Computation</b>                    | <b>123</b> |
| 10.1       | Verifiable Secret Sharing . . . . .                                | 124        |
| 10.2       | Efficient Error Correction . . . . .                               | 126        |

|   |            |
|---|------------|
| 10.3 Active-Adversary Secure MPC . . . . .          | 127        |
| 10.4 Active-Adversary MPC from AG Schemes . . . . . | 130        |
| <b>Nederlandse Samenvatting</b>                     | <b>141</b> |
| <b>Acknowledgements</b>                             | <b>143</b> |
| <b>Curriculum Vitae</b>                             | <b>145</b> |

## Part I

# Introduction



# Chapter 1

## History and Overview

This thesis deals with algebraic aspects of cryptography, with as main topics of interest secure message transmission and secure multi-party computation. The overview in this chapter provides the historical context for the research that is presented in this work.

### 1.1 Secure Communication

Historically, cryptography was exclusively concerned with securely communicating messages in the presence of an adversary. As an illustration, one can consider the historical setting where all message transmission was still being performed by couriers. Couriers carrying messages are typically vulnerable to interception by enemy forces, so extra means were necessary to ensure that capturing a courier would not automatically expose the contents of the message he was carrying. The approach was to ‘scramble’ any message in advance, in such a manner that no-one intercepting the message would be able to undo the scrambling operation and determine the contents of the original message. To make such a method work, the scrambling method needs to be reversible and known exclusively by the sender and the intended receiver of the message.

It is known that Julius Caesar used such a scrambling method to communicate sensitive messages. This fact is well-preserved due to a set of biographies on the lives of twelve successive Roman rulers called “*De Vita Caesarum*” written by Suetonius. The system used by Caesar is commonly referred to as the Caesar cipher and works as follows. Encryption (as scrambling is called nowadays) proceeds by shifting all letters in the message three positions forward in the alphabet, for example turning any occurrence of ‘A’ into a ‘D’ and the letter ‘Z’ into a ‘C’. To decrypt an encrypted message, the receiver just reverses the shifts. Key to the security of this system was

the fact that no outsider knew what technique was used to encrypt the messages, while it of course also helped that very few people knew how to read in the first place.

Encryption schemes have undergone an important conceptual improvement. In the early days the security of the message was achieved by keeping the encryption scheme private, and could be made substantially easier to break if any details about the scheme managed to leak out. This is a particularly important issue when the encryption method becomes automated, as is demonstrated by numerous examples of systems that were broken during the second world war after the other side managed to obtain one of the encryption devices. The improvement is that nowadays good encryption schemes are made public, but are keyed in such a way that every distinct possible key leads to a distinct encryption method. In a sense this corresponds to having a large range of encryption methods of which a random one is privately used to perform the actual encryption.

Applying this idea to the Caesar cipher, for example, one obtains a more flexible scheme when one considers the range of schemes corresponding with different numbers of shifts. Even when the shifting mechanism is made public, there are still 23 possible shifts that might have been applied, and if arbitrary letters have been added to the original text it can be quite some work to figure out which of these has been applied. Of course this particular trick is of fairly limited use now with the availability of high-powered computers, but it does demonstrate the general idea.

The idea of making the range of encryption methods public while only relying on the knowledge of the key for security is known as *Kerckhoffs' principle*, due to a publication on encryption written by the Dutch linguist and cryptographer Auguste Kerckhoffs in the 19<sup>th</sup> century. This approach has several advantages, as it usually leads to easier to analyze encryption schemes and allows to create (physical) implementations of encryption schemes, requiring a key to operate, that do not allow the scheme to be broken by reverse engineering the implementation.

In order to make an encryption scheme useful, it is necessary that the probability of guessing the correct key can be made arbitrarily small. This is particularly made difficult by the current state of the art of hardware, that allows to try out many keys in sequence in a short amount of time. Such hardware can at very high speed check whether a decryption under a certain key leads to a message that is likely to be the original message.

The obvious solution to bypass this problem is to make the set from which the key is chosen very large, which makes the chance of trying the correct key very small. As an extreme example, assume that we modify the Caesar cipher in such a way that every position in the text obtains its own unique shift, where we even allow the 'zero-shift' that leaves letters unchanged. If the shift for every position is chosen independently from the other positions and uniformly at random, the scheme becomes unbreakable as long as the chosen shifts remain completely private and the selected shifts are only

## 1.1. Secure Communication

---

used to encrypt once. This is easy to see, as every letter in the message can potentially map to any letter in the encrypted text without any correlation between the letters. This system, which is a variant of the *one-time-pad* cryptosystem, in fact provides perfect security. However, note that in order to use the scheme, the ‘key’ consisting of the specific shifts to be used has to be at least as long as the message that is to be transmitted, it needs to be exchanged before use and a new such key has to be used every time an encryption is required.

At the time of World War II, encryption systems had become more sophisticated. In particular, encryption and decryption procedures had become so complex that special equipment was required to perform these operations efficiently. Furthermore, the schemes were designed in such a way that secret keys could be reused with a limited decrease in security between uses. However, exchanging secret keys securely remained a sensitive problem.

In 1948, a couple of years after the end of the war, Shannon published a landmark paper founding information theory [67]. This theory formed a mathematical basis for the study of communication under the presence of noise or an interfering adversary. As a corollary of this theory it turned out that in order to encrypt a message with perfect security, the key needs to be at least as long as the message. This demonstrates that, when one requires perfect security, the one-time-pad is essentially optimal.

As first discovered in 1993 by Maurer [58], one can circumvent this restriction on the key-length by embracing non-perfect security, which allows for a small error probability. Essentially, the best one can hope for is an encryption system for which the error probability and the level of security scale with the length of the secret key. To alleviate the key-exchange problem, the key for such a scheme should additionally be reusable with only a negligible decrease in security between uses.

Another important problem in secure communication is *authentication*. Authentication provides a means to let a recipient of a message verify that the message has originated from a specific sender. Authentication is often established by techniques that let the sender add a message-dependent stamp to the message, called a *signature*. Note however that signatures typically have the stronger property that the originator of the message can be verified by anyone rather than just the recipient of the message.

In 1976, Diffie and Hellman [27] provide a solution to the seemingly paradoxical problem that a secret key needs to be communicated securely before the key can be used to establish secure communication. The solution they provide relies on the assumptions that the computational power of the adversary is bounded and that the parties involved are able to provide authentication. As such, it is in essence a technique to bootstrap message secrecy during communication from authentication.

In their work Diffie and Hellman additionally propose a new paradigm, called *public-key cryptography*. In public-key cryptography, every participant holds both a public key and a private key, where the private key is kept secret and the public

key is accessible by everyone. When anyone needs to privately send a message to a participating receiver he simply encrypts the message using the corresponding public key. The receiver is then able to decrypt the message using his matching private key, while the message remains private with respect to everyone else. Diffie and Hellman furthermore demonstrate a general method that allows public-key cryptosystems to be used to create signatures.

Shortly after Diffie and Hellman's publication, in 1978, Rivest, Shamir and Adleman [62] published the very first public-key encryption scheme, based on the hardness of factoring, which is now known under the name RSA. It is worthwhile to note that it has since been claimed that the idea of public-key encryption had already been developed by Ellis in 1969 and that a first version of the RSA protocol had already been discovered in 1974 by Cocks, both at the time working at the top-secret Government Communications Headquarters (GCHQ) in Cheltenham, Great Britain. Due to the classified nature of the establishment the work had been kept secret until 1997.

The solutions for encryption and authentication mentioned thus far all rely on at least one out of two assumptions, i.e., that there is a method available to exchange secret keys prior to communication and/or that certain computational restrictions on the adversary hold. There are two other lines of work that we only briefly mention here.

In the *bounded storage model*, which was coined by Cachin and Maurer [10], one assumes a given limit on the storage capacity of the adversary. In its simplest form, the model involves a random source broadcasting massive amounts of data, where the sender, receiver and adversary are listening in on the source. The sender and receiver agree on a small number of indices that indicate which elements broadcast by the source will be used to construct a secret key. Since the adversary does not know which elements are relevant, the best approach for an attack is to store as much of the broadcast data as possible with the goal of computing (part of) the secret key later. However, since the adversary is swamped with information, such an attack quickly becomes infeasible. It is worth noting that since the sender and receiver need only listen in on a small part of the broadcast, they only require very small storage capacity. Following this initial setting, the theory of communication in the bounded storage model has further advanced, and requires increasingly advanced information theory.

In the model of *quantum cryptography* one assumes the laws of quantum mechanics hold. One particularly useful aspect of quantum mechanics is that it is impossible to observe quantum states without disturbing them. When one designs quantum communication protocols correctly, this enables the sender and receiver to detect such disturbances caused by an eavesdropper, which allows them to achieve secure communication. Some quantum cryptography, such as the key-exchange system due to Bennett and Brassard [7], is already used in practice and in particular several im-

## 1.2. Secure Computation

---

plementations of quantum communication systems are already sold on the market. However, these systems still suffer from some technical limitations. For instance, they are not yet wireless and currently have a maximum achievable communication range. It is worth noting that the most important computational problems that are currently used for encryption, such as the hardness of factoring and the discrete log problem, are in principle efficiently solvable on a *quantum computer*. Although some very small-scale quantum computers already exist, allowing to store up to a maximum of ten qubits in memory, there still remains a lot of research to be done concerning the feasibility of large scale quantum computers. Damgård, Fehr, Salvail and Schaffner have recently published a paper considering the bounded quantum storage model [26], that allows to essentially combine the advantages of the two namesakes of the model when quantum states are communicated and the adversary is only able to store (without observing) a very limited number of intercepted states.

In 1993, Dolev, Dwork, Waarts and Yung [29] introduce a different assumption that does away with the need for computational assumptions or an a priori key-exchange. In the model they consider multiple disjoint communication channels are available, as opposed to the classical model where only one communication channel is assumed. The protocols for this model are then required to be secure as long as only a limited number of the channels is unsafe during the protocol execution, where the required security can be either perfect or statistical (i.e, allowing a small error probability). In both cases perfect privacy for the message is required, but in the statistical case an arbitrarily small probability is allowed for the case that the protocol fails to transmit the message. The work of Dolev et al. has since set off an entire line of research.

Although it is not currently commonplace in practice to employ multiple communication channels in order to boost security and such an approach does require additional communication, it is not an unreasonable assumption given the many types of communication channels that are already available now. For instance, anyone can nowadays make more or less simultaneous use of the Internet, a telephone connection, the post office and a private courier, where most of these types of communication are unlikely to cross each others path at a mutual vulnerable point along the way. More importantly, since protocols in this model achieve perfect security without reliance on any computational assumption, they are guaranteed to remain useful regardless of advancements in computational power or the potential future rise of quantum computers with large enough memory.

## 1.2 Secure Computation

In the previous section we described how research in secure message transmission allows to combat passive eavesdropping or active interference on one or more com-

munication channels between two transmitting parties. We for now assume that this problem has been resolved, providing either computational or information theoretical security, resulting in secure and authenticated channels between any two parties that wish to transmit messages to one another. This removes any concern communicating parties might have for an outside intruder.

A question one can also ask however, is how to provide security against an *inside* intruder, i.e., an adversary that takes control of some of the parties involved in a joint computation process. In this case the parties we have to defend against are an integral part of the computation and cannot be simply shut out unless it is certain they are corrupted. This is a fundamentally different problem from secure communication, where we try to prevent the adversary from taking part in the communication in the first place, and as such requires fundamentally different techniques. Research in secure computation considers this problem in its full generality.

To be more precise, secure computation takes place in the following setting. There are  $n$  parties  $p_1, p_2, \dots, p_n$ , usually referred to as *players*, that each hold their own respective private input vector  $\vec{y}_i$  consisting of a finite number of elements from a finite field  $\mathbb{F}_q$  for  $i = 1, 2, \dots, n$ . Furthermore, there is an adversary that can take control of some of the players in the network, which are then said to be *corrupted*. The adversary can be *passive*, only reading the information obtained by corrupted players, or *active*, additionally taking full control of the actions of the corrupted players. Informally stated, the goal of secure computation is for the players to determine the function value of some given function  $F$  applied to their inputs under the presence of the adversary, while keeping the inputs  $\vec{y}_1, \vec{y}_2, \dots, \vec{y}_n$  as private as possible and guaranteeing correctness of the output.

A better way to describe this functionality is by way of a Gedankenexperiment where the players have access to some incorruptible, fully trusted mediator. In this setting, all players can simply send their inputs to the mediator using a secure channel. The mediator then computes the function value and sends it back to all of the players. Clearly, the only way for a player to disturb such a computation is to not provide any input at all. Furthermore, no collusion of players learns anything besides their own inputs, the output and anything that can be deduced from these values. Secure computation attempts to emulate the mediator when it is not available.

The problem of secure computation was first introduced and solved in 1982 by Yao [78] for the two-party case, where at most one player can be corrupted by a passive adversary, assuming the hardness of factoring large integers. The technique used is known as *Yao's garbled circuit* and involves a computation circuit for the function  $F$  where all inputs and outputs in the circuit are encrypted. The idea is that one party, called sender, constructs the circuit and that the other party, called receiver, only receives decryption keys corresponding to the inputs of the sender and the receiver. This allows the receiver to decrypt keys corresponding to gate outputs

in the circuit in a step-wise fashion until he in the end learns the output of the circuit computed on the inputs. If needed, the sender and receiver can then reverse roles so that both learn the correct output.

In 1987, Goldreich, Micali and Wigderson [38] demonstrate that if players are in a communication network where the adversary can corrupt only a minority of the players, then the players can compute any function securely even if the adversary is active. We note that the adversary is assumed to be *static*, which means that he has to select all the players to control at the start of the computation. When the adversary can select additional players dynamically during the computation, the adversary is called *adaptive*. To achieve their result, Goldreich et al. need to use an intractability assumption, but the crucial conceptual aspect that allows Goldreich et al. to move from security against a passive adversary to security against an active adversary is the use of *zero-knowledge*.

A zero-knowledge protocol is a protocol that allows one party, the *prover*, to convince another party, the *verifier* that he holds a witness to an NP-statement without leaking anything related to the witness to the verifier. For instance, the witness might correspond with the factorization of a large composite number that is the product of two primes, a secret key, or a proof of some complex mathematical theorem.

Instead of making the witness public, the prover and verifier execute an interactive protocol based on a circuit verifying the witness, where the verifier sends the prover a series of randomized challenges that the prover should be able to answer if he indeed holds the witness. On the other hand, if the prover does not hold the witness, he can only answer a challenge correctly with some constant probability. Due to the randomization used for the challenges, the outcomes are statistically independent of the witness.

The concept of zero-knowledge protocols was first introduced in 1985 by Goldwasser, Micali and Rackoff [40]. In their article, they give a first example of a zero-knowledge protocol for a particular mathematical language related to the quadratic residuosity problem. Goldreich, Micali and Wigderson [39] extend the technique to all languages in the NP-complexity class by demonstrating a zero-knowledge proof technique for the NP-complete problem of Boolean circuit satisfiability. Goldreich et al. [38] then use these techniques to let players prove in zero-knowledge that they execute the steps of their multi-party computation protocol correctly. Canetti, Feige, Goldreich and Naor [12] later extended the multi-party computation result of Goldreich et al. to deal with adaptive adversaries.

The security level of the secure computation protocols above is computational, as the security of the protocols relies on some computational intractability assumption (for instance the computational difficulty of factoring large integers). It is also possible to design protocols that are secure without reliance on a computational intractability

assumption, which are called *unconditionally* secure. Unconditionally secure protocols can handle adversaries with arbitrary computing power and in particular remain secure against an adversary having access to the strongest possible quantum computer. Often unconditionally secure protocols involve a small error probability; when the error probability is nonzero these protocols are said to be *statistically* secure, otherwise they are called *perfectly* or *information-theoretically* secure.

In the communication model for the unconditionally secure protocols every pair of players is connected by an authenticated information-theoretically secure communication channel, where the communication-network is synchronized and the players have a broadcast channel available that allows to transmit a message to all players at once. This model is known as the *secure channels model* for secure computation. A broadcast channel is not always required as a separate assumption in the model, as Lamport, Shostak and Pease [52] have shown that one can be simulated when less than one-third of the players can be corrupted.

We now discuss the results on perfectly secure multi-party computation protocols, i.e., protocols that are unconditionally secure and have a zero-error probability. The remarkable fact that protocols exist that are secure without computational intractability assumptions was independently proven by Ben-Or, Goldwasser and Wigderson [4] and Chaum, Crépeau and Damgård [14]. Their results demonstrate that  $n$  players can compute any function perfectly securely against an adversary corrupting up to  $t$  players if and only if  $t < n/2$  when the adversary is passive and if and only if  $t < n/3$  when the adversary is active. In the more general setting where the set of corrupt players can be any arbitrary set in some predefined list, called the *adversary structure*, Hirt and Maurer [42] demonstrate that perfectly secure protocols exist when the adversary is passive (active) if and only if no two (respectively no three) sets in the adversary structure cover the whole player set. These properties on the adversary structure are known as the  $Q^{(2)}$  and  $Q^{(3)}$  property respectively.

Cramer, Damgård and Maurer [23] initiate a mathematical theory for secure computation by demonstrating that one can use any linear secret sharing scheme with a  $Q^{(2)}$  ( $Q^{(3)}$ ) adversary structure to enable perfectly secure multi-party computation against a passive (active) adversary. This is done by transforming any linear secret sharing scheme into a scheme that has an important structural property called *multiplicativity*, combined with a general technique that allows to build perfectly secure multi-party computation protocols against a passive adversary from multiplicative linear secret sharing schemes.

Similarly, Cramer et al. describe how to transform linear secret sharing schemes into schemes with a stronger structural property called *strong multiplicativity*. Such schemes are then in turn used to construct *verifiable* secret sharing schemes. Verifiable secret sharing schemes guarantee share consistency and unique reconstruction in the presence of an active adversary and are a key primitive used to construct

## 1.2. Secure Computation

---

perfectly secure multi-party computation protocols against an active adversary. Although constructing perfectly secure multi-party computation protocols against an active adversary from strongly multiplicative linear secret sharing schemes can be done efficiently, it remains an open question whether such an efficient construction exists from linear secret sharing schemes with a  $Q^{(3)}$  adversary structure in general. However, if one allows for an error probability and the adversary structure of the linear secret sharing scheme is  $Q^{(2)}$ , there indeed exists an efficient construction for such protocols.

Although Cramer, Damgård and Dziembowski [22] have shown that it is not possible in general to use *arbitrary* secret sharing schemes to construct secure multi-party computation protocols, *linear* secret sharing is now considered to be a fundamental primitive of any unconditionally secure multi-party computation protocol. Essentially, in these protocols secret sharing can be seen as a distributed encryption function on the inputs. Since any function can be represented by an arithmetic circuit over the inputs consisting of addition and multiplication gates, the specification of a secure multi-party computation protocol merely needs to include a description of the dynamic operations on the encryptions that correspond with addition and multiplication of the encrypted values. A secure multi-party computation then proceeds by recursively applying these operations to the encryptions while traversing over the arithmetic circuit.

More specifically, secret sharing schemes allow to distribute shares in a secret value among a number of players in such a way that certain subsets of players can determine the secret value by pooling their shares and certain subsets of players cannot obtain any information on the secret value based on their collective shares. Any subset of players that can determine the secret is said to be *accepted* and any subset of players that cannot obtain any information about the secret is said to be *rejected*.

Secret sharing schemes exist for any player set in which the accepted and rejected subsets do not overlap and are *monotonous*, i.e., any subset of players that contains an accepted set is also accepted and any subset of players that is contained in a rejected set is also rejected. This corresponds with the intuition that adding more shares cannot reduce the amount of information on the secret and removing shares cannot increase the amount of information on the secret.

Although perfectly secure multi-party computation protocols are theoretically efficient in terms of the amount of required communication and computation, they are fairly costly in practice. This is mainly due to the fact that general multi-party computation protocols operate on a gate-by-gate basis in the arithmetic circuit of the given function. Furthermore, it is not yet known which functions in this model can be computed efficiently using a constant number of rounds, although it is known that this is the case for algebraic formulae [2], functions in non-deterministic logspace and some related counting classes [32, 46, 47, 3]. Completing this classification is in fact

one of the big open problems in secure computation.

Franklin and Yung [34] make multi-party computation protocols practically more attractive by introducing a technique that allows to simultaneously compute a given function on many inputs at the price of a single function value computation. The cost reduction one achieves with this method is amortized, since it only occurs when multiple computations can be combined.

Hirt, Maurer and Przydatek [43] introduce *player elimination*, which allows to remove corrupted players during the run of a protocol. Part of the protocol needs to be rerun after every elimination, but since the protocol continues with less players the communication required is reduced throughout the remaining execution. It is important to note Hirt et al. do not assume a broadcast channel to be available and include the cost of simulating a broadcast to the communication complexity. In this model and using a proper distribution of the detection stages within the protocol, Hirt et al. achieve an amortized cost reduction in the total communication required for multi-party computation that is cubic in the number of players.

Chen and Cramer [15] show how the inherent structure of algebraic-geometric codes allows to use them to construct strongly multiplicative linear secret sharing schemes. By additionally selecting a suitable family of underlying curves these schemes can be defined over smaller, or even constant-sized fields. The idea is to select algebraic curves with a high ratio between the number of points on the curve and the genus of the curve, such as those designed by Garcia and Stichtenoth [36]. These curves can have a number of points that is much larger than that of the underlying finite field, which allows for a larger number of players in the secret sharing scheme. This is due to the fact that every player is required to be linked to a distinct point. Furthermore, since the ratio mentioned is high, the resulting scheme can achieve parameters that are comparable to those achieved in previous schemes where the finite field is required to grow with the number of players. Applying the results of Cramer, Damgård and Maurer [23] this leads to multi-party computation protocols where the communication consists of elements in a small or constant-sized field. This allows to reduce the communication required for secure multi-party computation by a logarithmic factor in the number of players compared to previous schemes.

## Applications of Secure Multi-Party Computation

By their general nature, secure multi-party computation techniques have many potential applications. To emphasize and give a general intuition of this fact, we now list two striking examples of recent applications, where one is theoretical and one is of a practical nature. Additionally, we briefly hint here at some promising potential future applications.

**Low-Communication Zero-Knowledge from MPC.** In their remarkable 2007

paper, Ishai, Kushilevitz, Ostrovsky and Sahai [48] describe a theoretical application for perfectly secure multi-party computation protocols. The authors construct a zero-knowledge protocol for the NP-complete problem of circuit satisfiability that asymptotically only requires  $O(1)$  communication for each gate in the circuit. This improves the previously known best communication complexity of  $O(k)$  for each gate due to Cramer and Damgård [20], where the tolerated error probability is  $2^{-k}$ , and is expected to be optimal.

The result is achieved due to introduction of the novel concept of constructing *zero-knowledge protocols* from perfectly secure multi-party computation protocols. In the paper of Ishai et al., the prover executes in his mind a perfectly secure multi-party computation where the witness is split up into inputs for the players and where the circuit is such that it outputs 1 if the witness is correct and 0 otherwise. After finishing the execution of the mental game, the prover then commits himself to the view of every player consisting of its input, used randomness, incoming messages and outgoing messages, where each commitment both binds the prover to the corresponding view while hiding it unless the commitment is opened. The verifier then selects a suitable number of players at random and is allowed to see their views. The privacy requirements of the multi-party computation protocol ensure that all other inputs remain private, so that the witness also remains private, while a dishonest prover is detected with some constant probability.

Since the zero-knowledge protocol involves sending views of players to the verifier, which includes all transmissions from and to the player, the protocol becomes more efficient as the underlying multi-party computation protocol becomes more efficient. The best results obtained in the paper use the latest state-of-the-art techniques in general multi-party computation including the low-communication multi-party computation techniques based on algebraic-geometric codes on algebraic curves due to Chen and Cramer [15].

**Secure Multi-Party Computation in Practice.** Even though most of the interest in multi-party computation has been theoretical, a large-scale real-life practical application of the multi-party computation techniques has recently appeared in Denmark [31]. Due to reduced support for sugar beet production from the EU, it became important to allocate production contracts to Danish beet farmers in the most cost-effective way. It was decided that the best method of distributing the contracts would be via a nation-wide double-auction, where the sugar buyers specify for a range of prices how much sugar they would like to buy at every possible price and the beet farmers specify how much they are willing to produce at every possible price. Based on this the market clearing price is determined, the price at which the demand equals the supply, and contracts are distributed based on the amounts listed for the market clearing price.

A problem with this approach is that the sugar beet farmers do not sufficiently trust Danisco to handle the auction alone. On the other hand, Danisco sometimes uses sugar beet contracts as security for farmer debts and therefore might not find it acceptable to let the farmer's union handle the auction. This type of problem, where trust between parties is an important obstacle, is a natural setting in which multi-party computation techniques can be applied.

In this case multi-party techniques could indeed be used to provide an elegant solution, where the introduction of a neutral third party allowed to handle the auction without giving any of the parties involved too much control. The first auction based on these techniques was held successfully in 2007.

**Promising Future Applications.** Benchmarking is a method for two or more companies to compare their operational statistics, thus giving an indication of how well they are doing compared to their competitors. Although benchmarking is becoming more prominent, the precise statistics that need to be compared are typically sensitive enough that they need to remain private. Therefore, benchmarking is currently handled by trusted commercial third parties that get to compare all the actual statistics under some stringent privacy conditions. Multi-party computation techniques would allow to remove the commercial third party, thus removing the attached hiring cost and removing the risk of information leakage.

Another natural application for multi-party computation is electronic voting. Several multi-party computation-based voting systems have already been developed in the past, and some initial experimentation with these systems has already occurred. For instance, in 2006 French citizens living abroad already had their first opportunity to vote electronically. As the availability of internet access steadily increases, more wide-spread use of electronic voting is likely to occur.

## 1.3 Contributions

The main contributions of this work concern the areas of secure message transmission and secure multi-party computation and are listed below.

### 1.3.1 Perfectly Secure Message Transmission

We study the perfectly secure message transmission problem introduced in 1992 by Dolev, Dwork, Waarts and Yung. Our main contributions with regard to this problem are two-fold. First, we present all relevant known bounds and main ideas from the literature in a historical overview, which as a result makes the literature on this topic significantly more accessible.

Second, we prove that the known lower bound on the minimum required communication overhead, in the setting where two communication phases are tolerated, is in fact achievable. Soon after the publication of this result in [1], it has been proven that one cannot break this lower bound by allowing additional communication phases. Thus, effectively our result solves the perfectly secure message transmission problem.

As an interesting twist, we furthermore demonstrate that there nevertheless is motivation to consider perfectly secure three-phase protocols. More precisely, we demonstrate that when the adversary is expected to remain passive most of the time, there exist perfectly secure three-phase protocols in settings where perfectly secure one-phase protocols do not exist that most of the time complete their transmission after a single communication phase.

#### 1.3.2 Perfectly Secure Multi-Party Computation

We initiate a theory for ramp schemes, which can be seen as a generalization of the concept of linear secret sharing scheme. After redefining (strong) multiplicativity for the ramp scheme setting, we then design various (strongly) multiplicative ramp schemes that circumvent a number of important limitations on previously known (strongly) multiplicative linear secret sharing schemes. In particular, we design families of ramp schemes for which the field size can be kept constant, as opposed to previous schemes where the field size needed to grow with the number of players.

To be more precise, we consider “almost-threshold” ramp schemes that, for two given thresholds  $t$  and  $r$ , reject all subsets of the player set of size at most  $t$  and accept all subsets of the player set of size at least  $r$ . These schemes are studied using two distinct technical approaches.

First, we demonstrate the existence of good multiplicative ramp schemes using general coding theory [17]. This entails a fundamentally new method of constructing multiplicative schemes, as these schemes are traditionally constructed based on polynomial evaluation. By linking our new approach with a Gilbert-Varshamov type theorem that allows to estimate the parameters of a randomly selected code, we demonstrate the existence of infinite families of multiplicative ramp schemes with near-optimal parameters that can be defined over constant-sized fields. This mimics the effect that Chen and Cramer achieve using algebraic geometry for strongly multiplicative schemes, but using much more accessible techniques. Furthermore, we give a general construction for high information rate ramp schemes from error correcting codes that can be seen to encapsulate all ramp schemes and propose two explicit methods for constructing high information rate ramp schemes for any fixed choice of parameters.

Second, we introduce a new class of strongly multiplicative ramp schemes based on techniques from algebraic geometry [21, 16]. These schemes in fact embody the second

class of almost-threshold strongly multiplicative ramp schemes known in the literature, with the first one being represented by the technically similar algebraic-geometric class of strongly multiplicative ramp schemes introduced by Chen and Cramer and its well-known classical special cases. Finding other classes of strongly multiplicative ramp schemes is still a wide open problem in secure multi-party computation.

We then conclude with a brief overview of the techniques that are needed to construct perfectly secure multi-party computation protocols from multiplicative ramp schemes. Since similar techniques are well-known for the special case of multiplicative linear secret sharing schemes, we mainly give the explicit details on how to construct protocols secure against a passive adversary here and restrict ourselves to a brief description of the additional techniques required in the more general setting of ramp schemes for the active case.

## Chapter 2

# Preliminaries

In this chapter we provide an overview of some of the basic theory relevant to the results in this thesis. The first part consists of some classical facts on coding theory and algebraic geometry, as well as some more recent advanced results originating from the interaction between algebraic geometry and coding theory. The second part consists of an overview of relevant definitions and results on secret sharing, as well as some more specialized results.

### 2.1 Basic Coding Theory

Let  $\mathbb{F}$  be a finite field.

DEFINITION 1. *The Hamming weight  $w_H(\vec{x})$  of a vector  $\vec{x} = (x_1, x_2, \dots, x_n) \in \mathbb{F}^n$  is the number of non-zero positions in  $\vec{x}$ , i.e.,*

$$w_H(\vec{x}) = \#\{i \mid x_i \neq 0\}.$$

DEFINITION 2. *The distance  $d(\vec{x}, \vec{y})$  between two vectors  $\vec{x}, \vec{y} \in \mathbb{F}^n$  is the number of positions in which  $\vec{x}$  and  $\vec{y}$  differ, i.e.,*

$$d(\vec{x}, \vec{y}) = w_H(\vec{x} - \vec{y}).$$

DEFINITION 3. *For a subspace  $\{\vec{0}\} \subsetneq C \subset \mathbb{F}^n$ , the minimum distance  $d_{\min}(C)$  is defined by*

$$d_{\min}(C) = \min_{\vec{x}, \vec{y} \in C: \vec{x} \neq \vec{y}} \{d(\vec{x}, \vec{y})\}.$$

When  $C = \{\vec{0}\}$ , we define  $d_{\min}(C) = n$ .

DEFINITION 4. The ball  $B(\vec{x}, d) \subset \mathbb{F}^n$  of radius  $d$  around  $\vec{x} \in \mathbb{F}^n$  is defined in the Hamming metric as

$$B(\vec{x}, d) = \{\vec{y} \in \mathbb{F}^n \mid d(\vec{x}, \vec{y}) \leq d\}.$$

Consider the ball  $B(\vec{c}, \tau) \subset \mathbb{F}^n$  of radius  $\tau = \lfloor (d_{\min}(C) - 1)/2 \rfloor$  around a codeword  $\vec{c} \in C$ . Then the intersection  $B \cap C = \{\vec{c}\}$  and therefore for any error vector  $\vec{e} \in \mathbb{F}^n$  with  $w_H(\vec{e}) = t$ , we can decode the codeword  $\vec{c}$  uniquely from the vector  $\vec{c} + \vec{e}$  if  $t \leq \tau$ . It is this property that allows codes to be used as the basis for reliable communication in the presence of noise. In general, finding an efficient decoding algorithm for an arbitrary code is a hard problem. In fact, decoding random codes is an NP-complete problem. One of the classical challenges in coding theory is to construct codes with many codewords and large minimum distance, and for which an efficient decoding algorithm exists.

DEFINITION 5. Consider the  $n$ -dimensional vector space  $\mathbb{F}_q^n$ . A code is a non-empty subset  $C \subset \mathbb{F}_q^n$ . If  $C$  is a subspace, then it is a linear code. For  $n, k, d \in \mathbb{Z}_{\geq 0}$ , an  $[n, k, d]_q$ -code is a linear code  $C \subset \mathbb{F}_q^n$  where  $k = \dim_{\mathbb{F}_q}(C)$  and  $d_{\min}(C) \geq d$ .

We often omit  $d$  and/or the finite field  $\mathbb{F}_q$  and use the shorthands  $[n, k, d]$ -code and  $[n, k]$ -code instead. Note that if  $C \neq \{\vec{0}\}$  is a linear code,

$$d_{\min}(C) = \min_{\vec{x}, \vec{y} \in C: \vec{x} \neq \vec{y}} \{d(\vec{x}, \vec{y})\} = \min_{\vec{x}, \vec{y} \in C: \vec{x} \neq \vec{y}} \{d(\vec{x} - \vec{y}, \vec{0})\} = \min_{\vec{c} \in C \setminus \{\vec{0}\}} \{w_H(\vec{c})\}.$$

DEFINITION 6. If  $C \subset \mathbb{F}^n$  is a code and  $\emptyset \neq A \subset \{1, \dots, n\}$ , then the projection map  $\Pi_A$  is defined by

$$\begin{aligned} \Pi_A : \mathbb{F}^n &\rightarrow \mathbb{F}^{|A|} \\ \vec{x} &\mapsto (x_i)_{i \in A} \end{aligned}$$

where  $\vec{x} = (x_1, x_2, \dots, x_n)$ . We use shorthand  $\vec{x}_A$  for  $\Pi_A(\vec{x})$ .

DEFINITION 7. If  $C \subset \mathbb{F}^n$  is a code and  $\emptyset \neq A \subset \{1, \dots, n\}$ , then

$$C_A = \{\vec{c}_A \mid \vec{c} \in C\}.$$

DEFINITION 8. The dual code  $C^\perp$  of a linear code  $C$  consists of all vectors  $\vec{c}^* \in \mathbb{F}^n$  such that  $\langle \vec{c}^*, \vec{c} \rangle = 0$  for all  $\vec{c} \in C$ , where  $\langle \cdot, \cdot \rangle$  denotes the standard scalar product. Whenever  $d$  is used to denote the minimum distance of  $C$ ,  $d^\perp$  is used to denote the minimum distance of  $C^\perp$ .

By elementary linear algebra, if  $C$  is an  $[n, k, d]$ -code, the dual code  $C^\perp$  is an  $[n, n - k, d^\perp]$ -code for some  $d^\perp \in \mathbb{Z}_{\geq 0}$ . Since codes are defined over finite fields, it is in fact possible that  $\{\vec{0}\} \subsetneq C \cap C^\perp$ .

## 2.1. Basic Coding Theory

---

DEFINITION 9. A linear code  $C$  with  $C = C^\perp$  is self-dual.

DEFINITION 10. Suppose  $|\mathbb{F}| > n$ . Let  $x_0, x_1, \dots, x_n \in \mathbb{F}$  be distinct elements and let  $0 \leq t < n$ . The Reed-Solomon code  $C \subset \mathbb{F}^{n+1}$  is defined as

$$C = \{(f(x_0), f(x_1), \dots, f(x_n)) \mid f \in \mathbb{F}[X], \deg(f) \leq t\},$$

where  $\deg(f)$  denotes the degree of the polynomial  $f$ .

Note that since any polynomial  $f(X) \in \mathbb{F}[X]$  of degree  $\leq t$  can have at most  $t$  zeroes, it holds that  $d_{\min}(C) \geq n + 1 - t$  for any  $[n + 1, t + 1]$  Reed-Solomon code  $C$ . By the Singleton bound (see Section 2.2)  $d_{\min}(C) \leq n + 1 - t$ , so that in fact equality holds.

PROPOSITION 1. The Reed-Solomon code is an  $[n + 1, t + 1, n + 1 - t]$ -code.

When using a Reed-Solomon code, efficient decoding of up to  $\tau = \lfloor (n - t)/2 \rfloor$  errors is possible using for instance the Berlekamp-Massey algorithm [55] or the Berlekamp-Welch algorithm [8].

By Lagrange interpolation, a polynomial  $g(X) \in \mathbb{F}[X]$  of degree at most  $t$  is uniquely determined by  $t + 1$  evaluations  $b_1 = g(a_1), b_2 = g(a_2), \dots, b_{t+1} = g(a_{t+1})$  for distinct  $a_1, \dots, a_n \in \mathbb{F}$ . In fact, it holds that

$$f(X) = \sum_{i=1}^{t+1} b_i L_i(X),$$

where

$$L_i(X) = \prod_{j \neq i} \frac{X - x_j}{x_i - x_j}$$

are such that  $L_i(x_i) = 1$  and  $L_i(x_j) = 0$  when  $i \neq j$ . These facts together with Proposition 1 imply the corollaries below, which will be useful in the sequel.

Let  $C$  be an  $[n + 1, t + 1]$  Reed-Solomon code and let  $\emptyset \neq A \subset \{0, 1, \dots, n\}$ .

COROLLARY 1. If  $|A| \geq t + 1$ , then the projection map  $\Pi_A : C \rightarrow C_A$  is bijective.

COROLLARY 2. If  $|A| \geq t + 1$ , then for each  $i \in \{0, 1, \dots, n\}$  there exists a linear map  $L_{A,i} : C_A \rightarrow \mathbb{F}$  such that  $\Pi_{\{i\}}(\vec{c}) = (L_{A,i} \circ \Pi_A)(\vec{c})$  for any  $\vec{c} \in C$ .

### A Variation on Reed-Solomon Codes

Let  $\mathbb{F}_q$  be a finite field with  $q > n$ , the values  $x_0, x_1, \dots, x_n \in \mathbb{F}_q$  be distinct and  $y \in \mathbb{F}_{q^{t+1}}$  be such that  $[\mathbb{F}_q(y) : \mathbb{F}_q] = t + 1$ . We can define the code  $C' \subset \mathbb{F}_{q^{t+1}}^{n+1}$  by

$$C' = \{(f(x_0), \dots, f(x_{i-1}), f(y), f(x_{i+1}), \dots, f(x_n)) \mid f \in \mathbb{F}_q[X], \deg(f) \leq t\}.$$

Let  $A = \{0, 1, \dots, n\} \setminus \{i\}$ . Then the vectors  $\vec{c}_A$  with  $\vec{c} \in C'$  form an  $[n, t + 1]$  Reed-Solomon code  $C \subset \mathbb{F}_q^n$ . In particular, there is a one-to-one correspondence between this code and the Reed-Solomon code where the evaluation  $f(y) \in \mathbb{F}_{q^{t+1}}$  is replaced with  $f(x_i) \in \mathbb{F}_q$ . For instance, this implies that any  $t + 1$  coefficients  $f(x_j)$  in a codeword  $\vec{c} \in C'$  determine it. As Lemma 1 below shows, there additionally exists a one-to-one correspondence between the evaluation  $f(y)$  in the  $i^{\text{th}}$  position and the polynomial  $f \in \mathbb{F}_q[X]$  corresponding to the codeword.

LEMMA 1. *Let  $\mathbb{F}_q$  be a finite field and let  $\alpha \in \mathbb{F}_{q^{t+1}}$  be such that  $\mathbb{F}_q(\alpha) = \mathbb{F}_{q^{t+1}}$ . Then the map*

$$\begin{aligned} \phi : \{f \in \mathbb{F}_q[X] : \deg(f) \leq t\} &\rightarrow \mathbb{F}_{q^{t+1}} \\ f &\mapsto f(\alpha) \end{aligned}$$

*is an isomorphism of  $\mathbb{F}_q$ -vector spaces.*

*Proof.* Since the dimensions on both sides are equal, it suffices to show that  $\phi$  is injective. Suppose  $g \in \mathbb{F}_q[X]$  has degree at most  $t$  and that  $g(\alpha) = 0$ . Since  $g \in \mathbb{F}_q[X]$ ,  $g$  must be a multiple of the minimal polynomial  $h$  of  $\alpha$  in  $\mathbb{F}_q[X]$ . However,  $\deg(h) = t + 1$ , so  $g$  must be the zero polynomial.  $\square$

We refer to this variant of the Reed-Solomon code as the  *$i$ -concentrated Reed-Solomon code* and use the notation  $f_\beta$  to denote the unique polynomial  $f \in \mathbb{F}_q[X]$  with  $\deg(f) \leq t$  for which  $f(y) = \beta \in \mathbb{F}_{q^{t+1}}$ .

## 2.2 Classical Bounds from Coding Theory

We now state some classical results on the achievable minimum distance and dimension that will be useful later on.

DEFINITION 11.  $A_q(n, d)$  is the maximum possible number of codewords that a code  $C \subset \mathbb{F}_q^n$  with minimum distance  $\geq d$  can contain, i.e.,

$$A_q(n, d) = \max_{C \subset \mathbb{F}_q^n : d_{\min}(C) \geq d} |C|.$$

Similarly,  $A_q^{\text{lin}}(n, d)$  denotes the maximum when we let  $C$  range over all linear codes  $C \subset \mathbb{F}_q^n$  with minimum distance  $\geq d$ .

The following upper bound holds.

THEOREM 1. (*Singleton Bound*)

$$A_q(n, d) \leq q^{n-d+1}.$$

## 2.2. Classical Bounds from Coding Theory

---

*Proof.* Let  $C \subset \mathbb{F}_q^n$  have minimum distance  $d$  and consider the code

$$C' = \{(c_d, c_{d+1}, \dots, c_n) \mid (c_1, c_2, \dots, c_n) \in C\}.$$

We argue that  $|C'| = |C|$ . If not, there exist two codewords  $\vec{c}_1, \vec{c}_2 \in C$  that have equal coefficients in the last  $n - d + 1$  positions. This implies that  $d(\vec{c}_1, \vec{c}_2) \leq d - 1$ , which contradicts the fact that the minimum distance of  $C$  is  $d$ .

Counting the number of possible codewords in  $C'$  it now follows that

$$|C| = |C'| \leq q^{n-d+1}.$$

□

**COROLLARY 3.** *For any  $[n, k, d]$ -code,  $k + d \leq n + 1$ .*

**DEFINITION 12.**  $V_q(n, d)$  denotes the volume of a ball of radius  $d$ , i.e.,

$$V_q(n, d) = \sum_{i=0}^d \binom{n}{i} (q-1)^i.$$

The following two theorems assert the existence of various codes.

**THEOREM 2.** (*General Gilbert-Varshamov Bound*) *For  $n, d \in \mathbb{Z}_{\geq 0}$  and  $q > 0$  a prime power,*

$$A_q(n, d) \geq \frac{q^n}{V_q(n, d-1)}.$$

*Proof.* Consider a maximal code  $C \subset \mathbb{F}_q^n$  with minimum distance  $d$ . Since  $C$  is maximal, the balls of radius  $d - 1$  around the codewords of  $C$  must cover the entire space  $\mathbb{F}_q^n$ , i.e.,

$$|C| \cdot V_q(n, d-1) \geq q^n.$$

This implies the bound. □

Surprisingly, the Gilbert-Varshamov bound still holds when we require the codes to be linear.

**THEOREM 3.** (*Gilbert-Varshamov Bound for Linear Codes*) *For  $n, d \in \mathbb{Z}_{\geq 0}$  and  $q > 0$  a prime power,*

$$A_q^{lin}(n, d) \geq \frac{q^n}{V_q(n, d-1)}.$$

*Proof.* We prove that if for  $k \in \mathbb{Z}_{\geq 0}$  it holds that

$$q^{k-1} < \frac{q^n}{V_q(n, d-1)}$$

then there exists an  $[n, k, d]_q$ -code. The theorem then follows by taking a maximal such  $k$ .

For  $k = 0$ , the code  $\{\vec{0}\} \subset \mathbb{F}_q^n$  is an  $[n, 0, n]_q$ -code and the statement holds trivially. Suppose  $k > 0$  and that there exists an  $[n, k - 1, d]$  code  $C'$ . Since  $q^{k-1} \cdot V_q(n, d - 1) < q^n$ , the balls of radius  $d - 1$  around the  $q^{k-1}$  codewords of  $C'$  do not cover the entire space  $\mathbb{F}_q^n$ . We can therefore select a vector  $\vec{x} \in \mathbb{F}_q^n \setminus C'$  such that  $d(\vec{x}, \vec{c}) \geq d$  for all  $\vec{c} \in C'$ .

Define the code  $C = C' + \vec{x} \cdot \mathbb{F}_q$ . Then the code  $C$  has dimension  $k$  and minimum distance at least  $d$ , since any non-zero vector  $\vec{y} \in C$  is of the form  $\vec{y} = \vec{c} + a\vec{x}$  with  $a \in \mathbb{F}_q$  and  $\vec{c} \in C'$  and we have that

$$w_H(\vec{y}) = w_H(a^{-1}\vec{y}) = w_H(a^{-1}\vec{c} + \vec{x}) = d(-a^{-1}\vec{c}, \vec{x}) \geq d$$

if  $a \neq 0$  and

$$w_H(\vec{y}) = w_H(\vec{c}) \geq d$$

if  $a = 0$ . □

It is often helpful to study the asymptotic behavior of families of codes as  $n$  increases. The parameters of interest for these families of codes are the information rate  $k/n$  and the relative minimum distance  $d/n$ . In the following we consider the supremum  $\alpha_q(\delta)$  of the information rates  $k/n$  that are achievable for linear codes over  $\mathbb{F}_q$  with relative minimum distance at least  $\delta$ .

We require the following definition.

**DEFINITION 13.** *Let  $\delta$  be such that  $0 < \delta < (q - 1)/q$ . The  $q$ -ary entropy function  $H_q$  is defined*

$$H_q(\delta) = \delta \log_q(q - 1) - \delta \log_q \delta - (1 - \delta) \log_q(1 - \delta).$$

We also mention the following lemma, which is a consequence of Stirling's formula for  $n!$ .

**LEMMA 2.** *For  $0 < \lambda < \frac{1}{2}$ ,*

$$\frac{2^{nH_2(\lambda)}}{\sqrt{8n\lambda(1-\lambda)}} \leq \sum_{k=0}^{\lambda n} \binom{n}{k} \leq 2^{nH_2(\lambda)}.$$

*Proof.* See [53], Chapter 10, §11, Corollary 9. □

The binary version of the following theorem is now a consequence of the Gilbert-Varshamov bound and Lemma 2. The  $q$ -ary variant additionally requires a more general version of Lemma 2 that is derived using similar techniques to those used in the proof of Theorem 26 in Chapter 7.

## 2.3. Algebraic Geometry Preliminaries

---

THEOREM 4. (*Asymptotic Gilbert-Varshamov Bound*) For  $0 < \delta < (q - 1)/q$ ,

$$\alpha_q(\delta) \geq 1 - H_q(\delta).$$

The Gilbert-Varshamov bound shows the existence of codes with certain parameters and while we do not make this statement precise here, it can in fact be shown that asymptotically “most” codes have parameters that are “close”. However, finding explicit constructions of codes that attain the Gilbert-Varshamov bound turns out to be very hard. Nevertheless, it is important to find explicit constructions of codes, because it is in general very inefficient to correct errors for arbitrary codes even when the parameters of the code are known.

## 2.3 Algebraic Geometry Preliminaries

In this section we list some basic facts from algebraic geometry. The theory described here is mainly used in Section 8.2 and can therefore be skipped in a first study of this work.

Please note that we follow the approach of Tsfasman and Vlăduț [75], which emphasizes on aspects of algebraic geometry that play an important role in algebraic geometric coding theory. For a more general treatment of the subject, we can recommend the work of Silverman [69].

Throughout this section we let  $\mathbb{F}$  be an algebraically closed field.

### 2.3.1 Projective spaces, topologies and varieties

The *projective space*  $\mathbb{P}^n(\mathbb{F})$  over  $\mathbb{F}$  of dimension  $n$  is the set of equivalence classes  $(a_1 : a_2 : \dots : a_{n+1})$  of non-zero vectors  $(a_1, a_2, \dots, a_{n+1}) \in \mathbb{F}^{n+1}$ , where

$$(b_1 : b_2 : \dots : b_{n+1}) = (a_1 : a_2 : \dots : a_{n+1})$$

if  $(b_1, b_2, \dots, b_{n+1}) = (\lambda a_1, \lambda a_2, \dots, \lambda a_{n+1})$  for some  $\lambda \in \mathbb{F}^*$ . When the theory does not depend on the specific choice of  $\mathbb{F}$ , we use the shorthand notation  $\mathbb{P}^n$ .

The Zariski topology on  $\mathbb{P}^n$  is now defined as follows. A subset  $A \subset \mathbb{P}^n$  is said to be *closed* if and only if there exist homogeneous polynomials

$$F_1, F_2, \dots, F_k \in \mathbb{F}[X_1, X_2, \dots, X_{n+1}]$$

such that

$$A = \{(a_1 : a_2 : \dots : a_{n+1}) \in \mathbb{P}^n \mid \forall i \in \{1, 2, \dots, k\} : F_i(a_1, a_2, \dots, a_{n+1}) = 0\}.$$

The complement  $\mathbb{P}^n \setminus A$  of a closed set  $A$  is said to be *open*. Similarly, a subset  $A' \subset \mathbb{P}^n$  is said to be closed in a subset  $S \subset \mathbb{P}^n$  if  $A' = A \cap S$  for a closed subset  $A \subset \mathbb{P}^n$  and a subset  $B \subset S$  is open in  $S$  if and only if  $S \setminus B$  is closed in  $S$ .

A *quasi-projective* set  $A$  is an open subset of a closed subset  $S \subset \mathbb{P}^n$ . It is said to be *irreducible* if there do not exist two non-empty closed subsets  $A_1, A_2 \subsetneq A$  such that  $A = A_1 \cup A_2$ . An irreducible quasi-projective set is called a *quasi-projective variety*. A quasi-projective variety  $A \subset \mathbb{P}^n$  that is closed in  $\mathbb{P}^n$  is said to be *projective*. Finally, the *dimension* of a quasi-projective variety  $A$  is the largest integer  $n$  such that there exists a strictly descending chain of quasi-projective varieties

$$A = A_0 \supsetneq A_1 \cdots \supsetneq A_n \neq \emptyset,$$

where  $A_i$  is closed in  $A_{i-1}$  for  $i = 1, 2, \dots, n$ . A one-dimensional quasi-projective variety is called a *curve*.

### 2.3.2 Rational functions, valuations and divisors

A *rational function* on a quasi-projective variety  $A \subset \mathbb{P}^n$  is a fraction

$$F/G \in \mathbb{F}(X_1, X_2, \dots, X_{n+1}),$$

where  $F, G \in \mathbb{F}[X_1, X_2, \dots, X_{n+1}]$  are two homogeneous polynomials of the same degree such that  $G(P) \neq 0$  for some  $P \in A$ . We can define an equivalence relation on these functions where two rational functions  $F/G$  and  $F'/G'$  are in the same equivalence class if and only if

$$FG' - F'G = 0$$

on  $A$ . The set of classes of rational functions on  $A$  under this equivalence relation is denoted by  $\mathbb{F}(A)$ . It can be shown that  $\mathbb{F}(A)$  is in fact a field with respect to the usual addition and multiplication operations.

A rational function  $f \in \mathbb{F}(A)$  is called *regular* at  $P \in A$  if it can be represented by some

$$F/G \in \mathbb{F}(X_1, X_2, \dots, X_{n+1})$$

with  $G(P) \neq 0$ . If  $f$  is regular for any  $P \in A$  then  $f$  is said to be regular on  $A$ . The set of regular functions on  $A$  is denoted  $\mathbb{F}[A]$ .

The set of rational functions that are regular at  $P \in A$  is denoted by  $\mathcal{O}_P$ . It is in fact a commutative ring with  $1 \in \mathcal{O}_P$  and contains a unique maximal ideal

$$m_P = \{f \in \mathcal{O}_P \mid f(P) = 0\}.$$

Note that this implies that  $\mathcal{O}_P/m_P$  is a field.

Let now  $A$  be a curve. A point  $P \in A$  is called non-singular if  $m_P$  is a principal ideal, i.e.,

$$m_P = t_P \cdot \mathcal{O}_P$$

### 2.3. Algebraic Geometry Preliminaries

---

for some  $t_P \in \mathbb{F}(A)$ . Such  $t_P$  is called a *local parameter* at  $P$  and in fact every non-zero element  $f \in \mathcal{O}_P$  can uniquely be written in the form  $f = u \cdot t_P^n$  with a unit  $u \in \mathcal{O}_P$  and  $n \in \mathbb{Z}$ . In other words, the local ring  $\mathcal{O}_P$  is a discrete valuation ring via the map

$$\nu_P : \mathcal{O}_P \rightarrow \mathbb{Z}_{\geq 0}$$

defined by  $\nu_P(f) = n$ , where the value of  $n$  is independent of the choice of local parameter  $t_P$ . This evaluation map  $\nu_P$  can be extended to all of  $\mathbb{F}(A)$  by setting for any  $f = g/h$  with  $g, h \in \mathcal{O}_P$  that  $\nu_P(f) := \nu_P(g) - \nu_P(h)$ . The curve  $A$  is called *smooth* (or non-singular) if every point  $P \in A$  is non-singular.

A (Weil) *divisor* on a smooth projective curve  $C$  is a formal sum

$$D = \sum_{P \in C} a_P \cdot P$$

with  $a_P \in \mathbb{Z}$  for which the support  $\text{supp}(D)$ , i.e., the set of points  $P$  for which  $a_P$  is nonzero, is finite. Given two divisors  $D = \sum_{P \in C} a_P \cdot P$  and  $D' = \sum_{P \in C} a'_P \cdot P$ , we say that  $D \geq D'$  if  $a_P \geq a'_P$  for all the points  $P$  on the curve. The degree  $\text{deg}(D)$  of a divisor  $D = \sum_{P \in C} a_P \cdot P$  is the sum of its coefficients, i.e.,

$$\text{deg}(D) = \sum_{P \in C} a_P.$$

The divisor  $(f)$  of a rational function  $f \in \mathbb{F}(C)$  on a smooth projective curve  $C$  is defined by

$$(f) = \sum_{P \in C} \nu_P(f) \cdot P.$$

Divisors of the form  $(f)$  for  $f \in \mathbb{F}(C)$  are called *principal* and it is a well-known fact that  $\text{deg}((f)) = 0$ . Two divisors  $D$  and  $D'$  are said to be *linearly equivalent* if there exists a rational function  $f \in \mathbb{F}(C)$  such that  $D = D' + (f)$ .

Let  $\{U_i\}$  be a finite open covering of a smooth projective curve  $C$  and let  $\{f_i\}$  be rational functions such that

$$f_j \cdot f_k^{-1} \in (\mathbb{F}[U_j \cap U_k])^*$$

for any  $f_j, f_k \in \{f_i\}$ . Then  $(\{U_i\}, \{f_i\})$  is called a *Cartier divisor*. The following theorem demonstrates the connection between Cartier divisors and Weil divisors.

**THEOREM 5.** *Let  $(\{U_i\}, \{f_i\})$  be a Cartier divisor. There exists a unique Weil divisor  $D = \sum_{P \in C} a_P \cdot P$  such that for all indices  $i$*

$$D|_{U_i} = (f_i)|_{U_i},$$

where  $D|_U = \sum_{P \in U} a_P \cdot P$ . Conversely, for any Weil divisor  $D$  on a smooth projective curve  $C$  there exists an open covering  $\{U_i\}$  of  $C$  and rational functions  $\{f_i\}$  such that for all indices  $i$

$$D|_{U_i} = (f_i)|_{U_i}$$

and

$$f_j \cdot f_k^{-1} \in (\mathbb{F}[U_j \cap U_k])^*$$

for any  $f_j, f_k \in \{f_i\}$ .

### 2.3.3 Rational differential forms

Let  $U$  be an open subset of a smooth projective curve  $C$ . For a point  $P$  on  $C$  and a rational function  $f \in \mathbb{F}[U]$ , the *differential*  $d_P f$  of  $f$  at  $P$  is defined as

$$d_P f := f - f(P) \in m_P/m_P^2.$$

Let  $\Phi[U]$  be the  $\mathbb{F}[U]$ -module consisting of the maps  $\phi$  that send each  $P \in U$  to some  $\phi(P) \in m_P/m_P^2$ . Then the set  $\Phi[U]$  includes all maps  $df$  defined by

$$(df)(P) = d_P f$$

for  $f \in \mathbb{F}[U]$ .

A map  $\phi \in \Phi[U]$  is called a *differential form* regular on  $U$  if and only if for any  $P \in U$  there exists an open  $V \subset U$  with  $P \in V$  such that  $\phi|_V$  is in the  $\mathbb{F}[V]$ -submodule of  $\Phi[V]$  generated by the  $df$  with  $f \in \mathbb{F}[V]$ . The  $\mathbb{F}[U]$ -module of differential forms regular on  $U$  is denoted by  $\Omega[U]$ .

If there exists an open subset  $U \subset C$  such that  $\omega \in \Omega[U]$ , then  $\omega$  is called a *rational differential form* on  $C$ . If for open  $U, U' \subset C$  and  $\omega \in \Omega[U]$ ,  $\omega' \in \Omega[U']$  it holds that

$$\omega|_{U \cap U'} = \omega'|_{U \cap U'}$$

we say that  $\omega$  and  $\omega'$  define the same rational differential form on  $C$ . The set of rational differential forms on  $C$  under this equivalence relation is denoted by  $\Omega(C)$ .

**THEOREM 6.** *The set  $\Omega(C)$  forms a one-dimensional  $\mathbb{F}(C)$ -vector space.*

For any  $P \in C$  there exists an open subset  $U \ni P$  of  $C$  such that

$$\Omega[U] = \mathbb{F}[U] \cdot dt,$$

where  $t$  is a local parameter at  $P$ . It follows that for every rational differential form  $0 \neq \omega \in \Omega$  there exists an open covering  $\{U_i\}$  of  $C$  such that

$$\omega|_{U_i} = f_i \cdot dt_i$$

### 2.3. Algebraic Geometry Preliminaries

---

for certain  $f_i \in \mathbb{F}(C)$  and such that  $t_i - t_i(P)$  is a local parameter for any  $P \in U_i$ . It follows that

$$f_i/f_j \in (\mathbb{F}[U_i \cap U_j])^*,$$

so that  $(\{U_i\}, \{f_i\})$  gives a Cartier divisor which is denoted by  $(\omega)$ . In the sequel, we identify the Cartier divisor  $(\omega)$  with the corresponding Weil divisor.

Every pair of differential forms  $0 \neq \omega, \omega' \in \Omega(C)$  gives rise to linearly equivalent divisors, i.e.,

$$(\omega') = (\omega) + (f)$$

for some  $f \in \mathbb{F}(C)$ . Any such divisor  $(\omega)$  defined by a differential form  $\omega$  is called a *canonical divisor*. For any canonical divisor  $K$ , we have that

$$\deg(K) = 2g - 2.$$

Let  $\omega \in \Omega(C)$  be such that  $\omega = f \cdot dt$  in an open neighborhood of  $P \in C$ , where  $f \in \mathbb{F}(C)$  and  $t$  is a local parameter at  $P$ . The map  $Res_P$  maps the differential form  $\omega$  to the coefficient  $a_{-1}$  in the Laurent power series representation of  $f$  around  $t$ , say  $f = \sum_{-M}^{\infty} a_i \cdot t^i$ . It can be shown that this value does not depend on the choice of  $t$ . Furthermore, if

$$(\omega) = \sum_{P \in C} a_P \cdot P,$$

it can be shown that  $Res_P(\omega) = 0$  if  $a_P \geq 0$  and  $Res_P(\omega) \neq 0$  if  $a_P = -1$ .

The following well-known theorem gives a relation for a differential form  $\omega$  between the evaluations  $Res_P(\omega)$  in all the points  $P$  on a smooth projective curve  $C$ . Note here that the maps  $Res_P$  only give non-zero evaluations in a finite number of points  $P$  for any differential form  $\omega \in \Omega(C)$ .

**THEOREM 7. (Residue Theorem)** For any  $\omega \in \Omega(C)$  we have

$$\sum_{P \in C} Res_P(\omega) = 0.$$

For any divisor  $D$ , the corresponding *Riemann-Roch space*  $L(D)$  is defined by

$$L(D) = \{f \in \mathbb{F}(C) \mid (f) + D \geq 0\} \cup \{0\}.$$

This is a vector space over  $\mathbb{F}$  and its dimension is denoted  $\ell(D)$ . For any canonical divisor  $K$  we have  $\ell(K) = g$ , and for any divisor  $D$  with  $\deg(D) < 0$  we have that  $\ell(D) = 0$ .

The following important theorem is frequently used to determine the dimension of Riemann-Roch spaces.

**THEOREM 8. (Riemann-Roch Theorem)** *Let  $K$  be a canonical divisor. For any divisor  $D$ ,*

$$\ell(D) = \ell(K - D) + \deg(D) - g + 1.$$

*In particular,*

$$\ell(D) = \deg(D) - g + 1$$

*when  $\deg(D) > 2g - 2$ .*

We furthermore for any divisor  $D$  define the space  $\Omega(D)$  by

$$\Omega(D) = \{\omega \in \Omega(C) \setminus \{0\} \mid (\omega) + D \geq 0\} \cup \{0\}.$$

There exists an isomorphism  $L(K + D) \simeq \Omega(D)$  via the map  $f \mapsto f\eta$ , where  $(\eta) = K$ . Thus, the dimension of  $\Omega(D)$  can be determined using the Riemann-Roch Theorem.

### 2.3.4 $\mathbb{F}_q$ -rationality

Let  $\mathbb{F}_q$  be a finite field,  $\overline{\mathbb{F}}_q$  be algebraically closed over  $\mathbb{F}_q$  and let  $P$  be a point on a smooth projective curve  $C \subset \mathbb{P}^n(\overline{\mathbb{F}}_q)$  defined over  $\mathbb{F}_q$ , i.e., defined by equations  $F_1 = F_2 = \dots = F_m = 0$  with homogeneous polynomials  $F_1, F_2, \dots, F_m \in \mathbb{F}_q[X_1, X_2, \dots, X_{n+1}]$ . We say that the point  $P$  is  $\mathbb{F}_q$ -rational if it has a representation  $(x_1 : x_2 : \dots : x_{n+1}) \in \mathbb{P}^n(\overline{\mathbb{F}}_q)$  with  $x_1, x_2, \dots, x_{n+1} \in \mathbb{F}_q$ . A divisor  $D = \sum_{P \in C} a_P \cdot P$  is said to be  $\mathbb{F}_q$ -rational if  $D = \sum_{P \in C} a_P \cdot f(P)$  for any  $f \in \text{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$ , where  $\text{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$  denotes the Galois group of  $\overline{\mathbb{F}}_q$  over  $\mathbb{F}_q$ . Note that such a divisor can have support outside of the  $\mathbb{F}_q$ -rational points on  $C$ .

We define  $\mathbb{F}_q$ -rational functions to be functions  $L \in \overline{\mathbb{F}}_q(X_1, X_2, \dots, X_{n+1})$  where  $L$  has a representation  $G/H$  for which  $G, H \in \mathbb{F}_q[X_1, X_2, \dots, X_n]$  are homogeneous polynomials of the same degree and for which  $H(P) \neq 0$  for some  $P \in C$ . Note that evaluating an  $\mathbb{F}_q$ -rational function in an  $\mathbb{F}_q$ -rational point results in a value in  $\mathbb{F}_q$ .

The Riemann-Roch space of an  $\mathbb{F}_q$ -rational divisor  $D$  on  $C$  admits a basis defined over  $\mathbb{F}_q$ , i.e., it is equal to the  $\overline{\mathbb{F}}_q$ -linear span of certain  $\mathbb{F}_q$ -rational functions  $L_1, L_2, \dots, L_{l(D)} \in \overline{\mathbb{F}}_q(X_1, X_2, \dots, X_{n+1})$ . This is a consequence of the fact that the Riemann-Roch space of an  $\mathbb{F}_q$ -rational divisor is invariant under operations of  $\text{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$  (see [69], page 40, Lemma 5.8.1). We often restrict ourselves to the  $\mathbb{F}_q$ -rational functions in such Riemann-Roch spaces.

### 2.3.5 Algebraic-Geometric Codes

We now define algebraic-geometric codes, which are also known as geometric Goppa codes. These codes strictly generalize the construction of Reed-Solomon codes described earlier.

## 2.4. Algebraic-Geometric Bounds from Coding Theory

---

Let  $\mathbb{F}_q$  be a finite field and  $\mathcal{C} \subset \mathbb{P}^n(\overline{\mathbb{F}}_q)$  a smooth projective curve with at least  $n$  distinct  $\mathbb{F}_q$ -rational points. Fix a set  $D = \{P_1, \dots, P_n\}$  of  $\mathbb{F}_q$ -rational points on the curve  $\mathcal{C}$  and an  $\mathbb{F}_q$ -rational divisor  $G$  with support disjoint from  $D$ . The corresponding algebraic-geometric code  $C(D, G)$  based on  $\mathcal{C}$ ,  $D$  and  $G$  is now

$$C(D, G) = \{(f(P_1), f(P_2), \dots, f(P_n)) \mid f \in L(G), f \text{ is } \mathbb{F}_q\text{-rational}\}.$$

Since the points in  $D$  and the rational functions  $f$  are  $\mathbb{F}_q$ -rational, it follows that  $C(D, G) \subset \mathbb{F}_q^n$ . Furthermore, since  $L(G)$  is a vector space it is easy to see that the code  $C(D, G)$  is linear. If we take the curve  $\mathcal{C}$  to be the projective line  $\mathbb{P}^1(\overline{\mathbb{F}}_q)$ , the code  $C(D, G)$  is a generalized Reed-Solomon code.

The following bound, which can for instance be found in [75], is known on the parameters of algebraic-geometric codes.

**THEOREM 9.** *Let  $\mathcal{C} \in \mathbb{P}^n(\overline{\mathbb{F}}_q)$  be a smooth, projective curve of genus  $g$  and let  $0 < \deg(G) = a \leq n = |D|$ , where  $D$  and  $G$  are as above. Then the code  $C(D, G)$  is an  $[n, k, d]$ -code with*

$$k \geq a - g + 1$$

and

$$d \geq n - a.$$

It can be shown that the code  $C(D, G)^\perp \subset \mathbb{F}_q^n$ , the dual of  $C(D, G)$ , is in fact an algebraic-geometric  $[n, k', d']$ -code, where

$$k' \geq n - a + g - 1$$

and

$$d' \geq a - 2g + 2.$$

## 2.4 Algebraic-Geometric Bounds from Coding Theory

Although the Gilbert-Varshamov bound was published in 1952 by Gilbert [37], it was not until 1970 that Goppa [41] published the first explicit construction of codes that can asymptotically attain this bound. These codes, known as (classical) Goppa codes, also allow for efficient decoding algorithms for error-correcting (see for instance [30, 63]).

It remained an open question whether codes exist with parameters that are strictly better than those minimally predicted by the Gilbert-Varshamov bound. In 1982, Tsfasman, Vlăduț and Zink [76] proved that such codes do indeed exist. We now give this result in some more detail.

Let  $\mathcal{C}$  be a smooth, projective curve defined over  $\mathbb{F}_q$ , let  $\#\mathcal{C}(\mathbb{F}_q)$  denote the number of  $\mathbb{F}_q$ -rational points on  $\mathcal{C}$  and  $g(\mathcal{C})$  be the genus of  $\mathcal{C}$ . Define the quantity

$$A(q) = \limsup_{g(\mathcal{C}) \rightarrow \infty} \#\mathcal{C}(\mathbb{F}_q)/g(\mathcal{C}),$$

where  $\mathcal{C}$  ranges over all smooth projective curves defined over  $\mathbb{F}_q$ . Furthermore, recall from Section 2.2 that the quantity  $\alpha_q(\delta)$  denotes the supremum of the information rates  $k/n$  that are achievable for linear codes over  $\mathbb{F}_q$  with relative minimum distance  $\delta$ .

We can now state the following bound due to Tsfasman, Vlăduț and Zink [76], which relies on the existence of function fields attaining the Drinfeld-Vlăduț bound in combination with the construction of algebraic-geometric codes from function fields.

**THEOREM 10.** (*Tsfasman-Vlăduț-Zink Bound*) *Let  $q$  be a prime power. Then*

$$\alpha_q(\delta) \geq 1 - A(q)^{-1} - \delta.$$

In order to interpret this bound one needs to have good lower bounds on the value of  $A(q)$ . The Drinfeld-Vlăduț bound states that  $A(q) \leq \sqrt{q} - 1$ . The exact value of  $A(q)$  is known when  $q$  is a square and was discovered independently by Ihara [45] and Tsfasman, Vlăduț and Zink [76]. In this case  $A(q) = \sqrt{q} - 1$  and the Drinfeld-Vlăduț bound is tight.

It follows that for  $q \geq 49$  a square, the Tsfasman-Vlăduț-Zink bound strictly improves upon the Gilbert-Varshamov bound. When  $q = \ell^3$  is a cube, we have  $A(q) \geq 2(\ell^2 - 1)/(\ell + 2)$ . For arbitrary  $q$  the known bound is much weaker; it is known that in this case  $A(q) > c \cdot \log q$  for some real constant  $c > 0$ .

It is worth noting that while the Tsfasman-Vlăduț-Zink bound proves the existence of codes with parameters that improve on the Gilbert-Varshamov bound, it does not show how to construct such codes. In 1996, Garcia and Stichtenoth [36] demonstrated the first explicit method to construct families of codes attaining the Tsfasman-Vlăduț-Zink bound. It has since been proven by Xing [77] in 2003 that if one allows non-linear code constructions one can actually go through the Tsfasman-Vlăduț-Zink bound.

## 2.5 Secret Sharing

In this section we list some known facts on secret sharing and emphasize on certain limitations in current-day secret sharing techniques that we aim to overcome in this work.

### 2.5.1 Definition

Our preferred method of defining secret sharing schemes uses information theory, which opens up a plethora of results that both allow one to argue very precisely about partial information leakage and to use a rather compact notation. Unfortunately, properly introducing basic information theory notions is beyond the scope of this work. To also accommodate the reader without a background in information theory we therefore provide two nearly equivalent definitions of secret sharing using two different theoretical perspectives.

We first define secret sharing schemes using elementary probability theory, for which we also briefly introduce some basic notions. We then, without theoretical background, provide a slightly more general definition based on information theory that generalizes the first approach in the sense that the secret and the shares can now all be drawn from different finite sets and that we do not require a uniform distribution for the secret.

#### Using Probability Theory

We introduce some notation for probability theory based on the book by Shoup [68]. Let  $\mathcal{U}$  be a finite set. A probability distribution  $(\mathcal{U}, P)$  specifies a probability function

$$P : \mathcal{U} \rightarrow \mathbb{R}_{\geq 0}$$

that has the property that

$$\sum_{u \in \mathcal{U}} P(u) = 1.$$

The elements of  $\mathcal{U}$  represent the possible outcomes of a random experiment, where the probability of outcome  $u \in \mathcal{U}$  is  $P(u)$ . An *event*  $\mathcal{E}$  is a subset of  $\mathcal{U}$ . The probability of  $\mathcal{E}$  is denoted  $P(\mathcal{E})$  and defined by

$$P(\mathcal{E}) = \sum_{e \in \mathcal{E}} P(e).$$

A *random variable*  $X$  is a function  $X : \mathcal{U} \rightarrow \mathcal{T}$ , where  $\mathcal{T}$  is some set, and we say that  $X$  *takes values in*  $\mathcal{T}$ . For  $t \in \mathcal{T}$ , “ $X = t$ ” denotes the event

$$\{u \in \mathcal{U} : X(u) = t\}.$$

This implies that

$$P(X = t) = \sum_{u \in X^{-1}(t)} P(u).$$

Given any function  $f : \mathcal{T} \rightarrow \mathcal{V}$ , where  $\mathcal{V}$  is some set, it now follows that  $f \circ X$  defines a new random variable that takes values in  $\mathcal{V}$ .

For an event  $\mathcal{F} \subset \mathcal{U}$  with  $P(\mathcal{F}) \neq 0$  and  $u \in \mathcal{U}$  we define

$$P(u \mid \mathcal{F}) = \begin{cases} P(u)/P(\mathcal{F}) & \text{if } u \in \mathcal{F} \\ 0 & \text{otherwise.} \end{cases}$$

For an event  $\mathcal{E} \subset \mathcal{U}$  we furthermore define

$$P(\mathcal{E} \mid \mathcal{F}) = \sum_{e \in \mathcal{E}} P(e \mid \mathcal{F}).$$

When we have multiple events  $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$  we often use the notation

$$P(\mathcal{E} \mid \{\mathcal{F}\}_{i=1}^n)$$

to denote

$$P\left(\mathcal{E} \mid \bigcup_{i=1}^n \mathcal{F}_i\right),$$

where  $P(\mathcal{E} \mid \emptyset)$  is defined to be  $P(\mathcal{E})$ . Furthermore, to simplify our definitions we use the convention that  $P(\emptyset) = 1$ .

We are now ready to give a definition of secret sharing based on probability theory.

**DEFINITION 14.** (*Secret Sharing*) Let  $\mathcal{U}$  be a finite set and let  $(\mathcal{U}, P)$  define a probability distribution. A secret sharing scheme consists of random variables  $S_0, S_1, \dots, S_n$  that take value in some finite set  $\mathcal{S}$  with  $|\mathcal{S}| \geq 2$  where

$$P(S_0 = s_0) = 1/|\mathcal{S}|$$

for all  $s_0 \in \mathcal{S}$ .

Let  $\mathcal{N}$  be the set  $\{1, 2, \dots, n\}$  and let  $2^{\mathcal{N}}$  denote the power set consisting of all possible subsets of the set  $\mathcal{N}$ . For any secret sharing scheme one can define two sets  $\Gamma, \mathcal{A} \subset 2^{\mathcal{N}}$ , where for a set  $B \subset \mathcal{N}$

- $B \in \Gamma$  if and only if for all  $s_1, s_2, \dots, s_n \in \mathcal{S}$  with

$$P(\{S_i = s_i\}_{i \in B}) \neq 0$$

the following holds:

$$\exists s_0 \in \mathcal{S} : P(S_0 = s_0 \mid \{S_i = s_i\}_{i \in B}) = 1.$$

- $B \in \mathcal{A}$  if and only if for all  $s_0, s_1, \dots, s_n \in \mathcal{S}$  with

$$P(\{S_i = s_i\}_{i \in B}) \neq 0$$

the following holds:

$$P(S_0 = s_0 \mid \{S_i = s_i\}_{i \in B}) = 1/|\mathcal{S}|$$

## 2.5. Secret Sharing

---

The sets in  $\Gamma$  are said to be *accepted* by the secret sharing scheme, while the sets in  $\mathcal{A}$  are said to be *rejected*.

The values taken on by  $S_0$  are called *secrets* and the values taken on by  $S_i$  for  $i \in \{1, 2, \dots, n\}$  are called *shares*. We note that the definition can be made more general by allowing the variables  $S_0, S_1, \dots, S_n$  to take value in different finite sets  $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_n$ .

The sets  $\Gamma$  and  $\mathcal{A}$  are *monotonous*, in the sense that if  $B \in \Gamma$  and  $B \subset C \subset \mathcal{N}$  then also  $C \in \Gamma$  and that if  $D \in \mathcal{A}$  and  $E \subset D \subset \mathcal{N}$  then also  $E \in \mathcal{A}$ . It is easy to verify that  $\Gamma \cap \mathcal{A} = \emptyset$ . In general it is possible that for a secret sharing scheme there exist sets  $B \subset \mathcal{N}$  that are neither an element of  $\Gamma$  nor of  $\mathcal{A}$ .

DEFINITION 15. *If  $\Gamma = 2^{\mathcal{N}} \setminus \mathcal{A}$ , then the secret sharing scheme is perfect.*

DEFINITION 16. *A set  $B \in \Gamma$  is said to be a minimal accepted set if for any  $i \in B$  the set  $B \setminus \{i\}$  is not a member of  $\Gamma$ . A maximal rejected set is a set  $B \in \mathcal{A}$  such that for any  $j \in \mathcal{N} \setminus B$  the set  $B \cup \{j\}$  is not in  $\mathcal{A}$ .*

DEFINITION 17. *If an index  $i \in \mathcal{N}$  is not a member of any minimal accepted set  $B \in \Gamma$ , then  $i$  is said to be a dummy.*

DEFINITION 18. *A perfect secret sharing scheme that has no dummy indices and for which the variables  $S_0, S_1, \dots, S_n$  all take value in the same set  $\mathcal{S}$  is ideal.*

We will mainly be interested in secret sharing schemes without dummy indices, as shares corresponding with dummy indices are never required to determine the secret.

THEOREM 11. *For a perfect secret sharing scheme without dummy indices where the random variables  $S_0, S_1, \dots, S_n$  take value in the respective sets  $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_n$  with  $|\mathcal{S}_0| \geq 2$ , it holds that*

$$|\mathcal{S}_i| \geq |\mathcal{S}_0|$$

for every  $i \in \mathcal{N}$ .

PROOF. Let  $i \in \mathcal{N}$  and let  $B \subset \mathcal{N} \setminus \{i\}$  be a maximal rejected set. We assume that  $B \neq \emptyset$ , but the case  $B = \emptyset$  has a similar proof. Let  $\{s_\ell\}_{\ell \in B}$  be such that

$$P(\{S_\ell = s_\ell\}_{\ell \in B}) \neq 0.$$

Then in particular for any  $s_0 \in \mathcal{S}_0$

$$P(S_0 = s_0 \mid \{S_\ell = s_\ell\}_{\ell \in B}) = 1/|\mathcal{S}_0| > 0.$$

Since the set  $B \cup \{i\}$  is accepted, this implies that the value  $s_i$  determines  $s_0$  when  $\{s_\ell\}_{\ell \in B}$  are given. Every  $s_0 \in \mathcal{S}_0$  has non-zero probability given the shares  $\{s_\ell\}_{\ell \in B}$ , and therefore for every  $s_0 \in \mathcal{S}_0$  there exists at least one  $s_i \in \mathcal{S}_i$  such that

$$P(S_0 = s_0 \mid \{S_\ell = s_\ell\}_{\ell \in B \cup \{i\}}) = 1.$$

This implies that  $|\mathcal{S}_i| \geq |\mathcal{S}_0|$ . △

### Using Information Theory

We now assume the reader has some background knowledge of information theory and present a general definition of secret sharing in terms of information theory. A good reference for the information theory required is the book of Cover and Thomas [18].

In this section we let  $H$  denote the standard entropy function from information theory and use the notation  $H(\cdot|\cdot)$  to denote the standard conditional entropy function.

**DEFINITION 19.** (*Secret Sharing*) Let  $\mathcal{U}$  be a finite set and let  $(\mathcal{U}, P)$  define a probability distribution. A secret sharing scheme consists of random variables  $S_0, S_1, \dots, S_n$  that take value in some respective finite sets  $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_n$  where

$$H(S_0) > 0.$$

For any subset  $B \subset \mathcal{N}$  we use the notation  $S_B$  as shorthand for  $(S_i)_{i \in B}$ . One can again define the two sets  $\Gamma, \mathcal{A} \subset 2^{\mathcal{N}}$  for a secret sharing scheme, where for a set  $B \subset \mathcal{N}$

- $B \in \Gamma$  if and only

$$H(S_0|S_B) = 0.$$

- $B \in \mathcal{A}$  if and only if

$$H(S_0|S_B) = H(S_0),$$

where we define  $H(S_0|S_B) = H(S_0)$  when  $B = \emptyset$ .

The sets in  $\Gamma$  are again said to be *accepted* by the secret sharing scheme, while the sets in  $\mathcal{A}$  are again said to be *rejected*.

Definitions 15 to 18 carry over to the secret sharing schemes defined in this section. It is now clear which sets  $B \subset \mathcal{N}$  are neither in  $\Gamma$  nor in  $\mathcal{A}$ ; namely exactly those sets for which

$$0 < H(S_0|S_B) < H(S_0).$$

These sets are said to have *partial information*.

The following theorem can be seen as an improved variant of Theorem 11 and was proven in 1991 by Capocelli, De Santis, Gargano and Vaccaro [13], following a special case by Karnin, Greene and Hellman [50] proved in 1983.

**THEOREM 12.** For a perfect secret sharing scheme without dummy indices where the random variables  $S_0, S_1, \dots, S_n$  take value in the respective sets  $\mathcal{S}_0, \mathcal{S}_1, \dots, \mathcal{S}_n$ , it holds that

$$H(S_i) \geq H(S_0).$$

for  $i = 1, 2, \dots, n$ .

## 2.5. Secret Sharing

---

PROOF. Let  $i \in \mathcal{N}$  and let  $B \subset \mathcal{N} \setminus \{i\}$  be a maximal rejected set. We assume for simplicity that  $B \neq \emptyset$ . Then  $H(S_0|S_B) = H(S_0)$  and  $H(S_0|S_B S_i) = H(S_0|S_{B \cup \{i\}}) = 0$ .

Recall that for any random variables  $X, Y$  it holds that  $H(X|Y) \geq 0$  and recall the formula for (conditional) mutual information  $I(X; Y|Z)$  for any random variables  $X, Y, Z$ :

$$I(X; Y|Z) = H(X|Z) - H(X|YZ) = H(Y|Z) - H(Y|XZ) \geq 0.$$

We can now deduce the statement of the theorem:

$$\begin{aligned} H(S_i) &= H(S_i) - H(S_i|S_B) + H(S_i|S_0 S_B) + H(S_i|S_B) - H(S_i|S_0 S_B) \\ &= I(S_i; S_B) + H(S_i|S_0 S_B) + I(S_0; S_i|S_B) \\ &= I(S_i; S_B) + H(S_i|S_0 S_B) + H(S_0|S_B) - H(S_0|S_{B \cup \{i\}}) \\ &= I(S_i; S_B) + H(S_i|S_0 S_B) + H(S_0) \\ &\geq H(S_0) \end{aligned}$$

△

This theorem is important due to the relation between the entropy  $H(S)$  of a variable  $S$  that takes values in some finite set  $\mathcal{S}$  and the average number of bits that is required to encode an element in the image of  $S$  when using an optimal prefix-free encoding. Basically, using *Huffman codes* one can find an optimal prefix-free encoding for elements in the image of  $S$  (see [18]), where the average length of an encoding  $\ell$  is such that

$$H(S) \leq \ell < H(S) + 1.$$

Theorem 12 now has the following important consequence.

**COROLLARY 4.** *For any perfect secret sharing scheme without dummy indices, the average description length of any share is greater than or equal to the average description length of the secret.*

### 2.5.2 Linear Secret Sharing

The concept of linear secret sharing schemes has been developed through a number of publications, including work of Brickell [9], Stinson [73] and Karchmer and Wigderson [49]. In this section we give an overview of results on linear secret sharing based on a paper of Cramer, Damgård and Maurer [23].

Consider a monotone Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ . If we identify the input vector with a subset  $B \subset \mathcal{N}$ , by setting the  $i^{\text{th}}$  input bit to 1 if and only if  $i \in B$ , we can define the rejected sets to be those sets  $B \subset \mathcal{N}$  for which  $f(B) = 0$

and the accepted sets to be those for which  $f(B) = 1$ . We call the set  $\mathcal{A}_f$  of rejected sets the *adversary structure* and the set  $\Gamma_f$  of accepted sets the *access structure*. For an access structure  $\Gamma$ , the *dual*  $\Gamma^*$  is the set  $\{C \mid \mathcal{N} \setminus B \notin \Gamma\}$ , where it holds that  $(\Gamma^*)^* = \Gamma$ .

DEFINITION 20. A linear secret sharing scheme is a triple  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$ , where  $\mathbb{F}_q$  is a finite field,  $M \in \mathbb{F}_q^{(d+1) \times e}$  is a matrix with as its first row the unit vector  $\vec{e}_1 \in \mathbb{F}_q^e$  and  $\phi : \{1, 2, \dots, d\} \rightarrow \mathcal{N}$  is a surjective function. The size of  $(\mathbb{F}_q, M, \psi)$  is  $d$ .

Label the rows of the matrix by  $0, 1, 2, \dots, d$  and let  $M_i$  denote the  $i^{\text{th}}$  row of  $M$ . For any subset  $B \subset \mathcal{N}$ , we let  $M_B$  denote the submatrix consisting of the rows  $\{M_i\}_{i \in \psi^{-1}(B)}$ . Furthermore, let  $\text{Im}(M_B^T)$  denote the  $\mathbb{F}_q$ -linear span of the rows of  $M_B$  and  $\text{Ker}(M_B)$  consist of the vectors  $\vec{\kappa} \in \mathbb{F}_q^e$  such that  $M_B \cdot \vec{\kappa} = \vec{0}$ .

DEFINITION 21. For a linear secret sharing scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$

- a set  $B \subset \mathcal{N}$  is accepted if  $\vec{e}_1 \in \text{Im}(M_B^T)$ .
- a set  $B \subset \mathcal{N}$  is rejected if  $\vec{e}_1 \notin \text{Im}(M_B^T)$ .

If for a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  the set of rejected sets of  $\mathcal{M}$  is  $\mathcal{A}_f$  and the set of accepted sets of  $\mathcal{M}$  is  $\Gamma_f$ , then  $\mathcal{M}$  computes  $f$ .

A linear secret sharing scheme  $(\mathbb{F}_q, M, \psi)$  can be used to define a secret sharing scheme. Let  $(\mathcal{U}, P)$  define a probability distribution and let  $S$  be a random variable taking values in  $\mathbb{F}_q^e$  with equal probability. Define the random variable  $S_0 = M_0 \cdot S$  taking values in  $\mathbb{F}_q$  and the random variables  $S_i = M_{\{i\}} \cdot S$  taking values in  $\mathbb{F}_q^{d_i}$  with  $d_i = |\psi^{-1}(i)|$  for  $i = 1, 2, \dots, n$ .

THEOREM 13. The variables  $S_0, S_1, \dots, S_n$  define a secret sharing scheme, where the accepted and rejected sets coincide with those of the linear secret sharing scheme.

*Proof.* First note that  $\text{Im}(M_B^T) = \text{Ker}(M_B)^\perp$ , which follows from a basic linear algebra argument. Therefore, we have that  $\vec{e}_1 \notin \text{Im}(M_B^T)$  if and only if there exists an element  $\vec{\kappa} \in \mathbb{F}_q^e$  such that  $M_B \cdot \vec{\kappa} = \vec{0}$  and the first coordinate of  $\kappa$  is nonzero.

Assume a set  $B \subset \mathcal{N}$  is accepted, i.e.,  $\vec{e}_1 \in \text{Im}(M_B^T)$ . Then there exists a vector  $\vec{\lambda} \in \mathbb{F}_q^{|B|}$  such that  $M_B^T \vec{\lambda} = \vec{e}_1$ . Since  $S_0 = \langle \vec{e}_1, S \rangle$  it now follows that

$$\begin{aligned} S_0 &= \langle \vec{e}_1, S \rangle \\ &= \langle M_B^T \cdot \vec{\lambda}, S \rangle \\ &= \langle \vec{\lambda}, M_B \cdot S \rangle. \end{aligned}$$

Here  $\vec{\lambda}$  only depends on the matrix  $M_B$ , so the random variable  $S_0$  can in fact be defined via a linear function on the variables  $\{S_i\}_{i \in B}$ . This implies that  $H(S_0 \mid S_B) = 0$  and the set  $B$  is accepted according to Definition 19.

## 2.5. Secret Sharing

---

Now assume a set  $B \subset \mathcal{N}$  is rejected. Then there exists an element  $\vec{\kappa} \in \mathbb{F}_q^e$  such that  $M_B \cdot \vec{\kappa} = \vec{0}$  and the first coordinate of  $\kappa$  is nonzero. We assume without loss of generality that the first coordinate of  $\kappa$  is 1. This implies that for any  $\delta \in \mathbb{F}_q$ ,

$$\begin{aligned} M_B \cdot (S + \delta \vec{\kappa}) &= M_B \cdot S + \delta \cdot (M_B \cdot \vec{\kappa}) \\ &= M_B \cdot S + \delta \cdot \vec{0} \\ &= M_B \cdot S, \end{aligned}$$

while

$$\begin{aligned} M_0 \cdot (S + \delta \vec{\kappa}) &= M_0 \cdot S + \delta \cdot (M_0 \cdot \vec{\kappa}) \\ &= M_0 \cdot S + \delta \cdot (\epsilon_1 \cdot \vec{\kappa}) \\ &= M_0 \cdot S + \delta \end{aligned}$$

This implies that the random variable  $S_0$  is statistically independent of the random variables  $\{S_i\}_{i \in B}$ , i.e.,  $H(S_0 \mid S_B) = H(S_0)$  and the set  $B$  is rejected according to Definition 19.  $\square$

### 2.5.3 Multiplicative Linear Secret Sharing Schemes

We now describe two structural properties for linear secret sharing schemes that are important for the application of linear secret sharing schemes in secure multi-party computation. These properties are called *multiplicativity* and *strong multiplicativity* and were first introduced by Cramer, Dangård and Maurer [23].

For any two vectors  $\vec{x} = (x_1, x_2, \dots, x_e), \vec{y} = (y_1, y_2, \dots, y_e) \in \mathbb{F}_q^e$ , let  $\vec{x} \otimes \vec{y}$  denote the vector

$$(x_1 \cdot \vec{y}, x_2 \cdot \vec{y}, \dots, x_n \cdot \vec{y}) = (x_1 y_1, x_1 y_2, \dots, x_e y_{e-1}, x_e y_e) \in \mathbb{F}_q^{e^2}.$$

Let  $V_i \subset \mathbb{F}_q^e$  denote the subspace spanned by the row vectors of the matrix  $M_{\{i\}}$  and  $\hat{V}_i$  denote the subspace  $V_i \otimes V_i \subset \mathbb{F}_q^{e^2}$  spanned by all vectors  $\vec{x} \otimes \vec{y}$  with  $\vec{x}, \vec{y} \in V_i$ . Furthermore, let  $\hat{V}_B$  denote the subspace of  $\mathbb{F}_q^{e^2}$  spanned by all vectors in the spaces  $\{\hat{V}_i\}_{i \in B}$ .

DEFINITION 22. A linear secret sharing scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$  is multiplicative if

$$\vec{\epsilon}_1 \otimes \vec{\epsilon}_1 \in \hat{V}_{\mathcal{N}}.$$

DEFINITION 23. A linear secret sharing scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$  is  $\mathcal{A}$ -strongly multiplicative if the following two conditions hold.

1.  $\mathcal{M}$  rejects all sets in  $\mathcal{A}$ .

2. For any set  $B \in \mathcal{A}$ ,  $\mathcal{M}$  is multiplicative with respect to the set  $C = \mathcal{N} \setminus B$ , i.e.,

$$\vec{\epsilon}_1 \otimes \vec{\epsilon}_1 \in \hat{V}_C.$$

If the linear secret sharing scheme is ideal, i.e.,  $d = n$  and the map  $\psi$  is a bijection, we can replace Definitions 22 and 23 with the following convenient equivalent definitions. In fact, these definitions exactly describe how one uses multiplication properties in practice.

DEFINITION 24. An ideal linear secret sharing scheme  $\mathcal{M}$  is multiplicative if there exist  $\lambda_1, \dots, \lambda_n \in \mathbb{F}_q$  such that for any two secrets  $s, s'$  with respective shares  $s_1, s_2, \dots, s_n$  and  $s'_1, s'_2, \dots, s'_n$  we have that

$$s \cdot s' = \sum_{i=1}^n \lambda_i s_i s'_i.$$

DEFINITION 25. An ideal linear secret sharing scheme  $\mathcal{M}$  is  $\mathcal{A}$ -strongly multiplicative if

1.  $\mathcal{M}$  rejects all sets in  $\mathcal{A}$ .
2. For any set  $B \in \mathcal{A}$ ,  $\mathcal{M}$  is multiplicative with respect to the set  $C = \mathcal{N} \setminus B$ , i.e., given any set  $B \in \mathcal{A}$  there exist  $\{\lambda_i\}_{i \in C}$  in  $\mathbb{F}_q$  such that for any two secrets  $s, s'$  with respective shares  $s_1, s_2, \dots, s_n$  and  $s'_1, s'_2, \dots, s'_n$  it holds that

$$s \cdot s' = \sum_{i \in C} \lambda_i s_i s'_i.$$

DEFINITION 26. An adversary structure  $\mathcal{A}$  is  $Q^{(2)}$  if there are no two sets  $B_1, B_2 \in \mathcal{A}$  such that  $B_1 \cup B_2 = \mathcal{N}$ . Similarly, an adversary structure  $\mathcal{A}$  is  $Q^{(3)}$  if there are no three sets  $B_1, B_2, B_3 \in \mathcal{A}$  such that  $B_1 \cup B_2 \cup B_3 = \mathcal{N}$ . A monotone Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is  $Q^{(2)}$  (respectively  $Q^{(3)}$ ) if and only if  $\mathcal{A}_f$  is  $Q^{(2)}$  (respectively  $Q^{(3)}$ ).

Cramer et al. prove that if a linear secret sharing scheme with adversary structure  $\mathcal{A}$  is multiplicative (respectively  $\mathcal{A}$ -strongly multiplicative) then  $\mathcal{A}$  is  $Q^{(2)}$  (respectively  $Q^{(3)}$ ).

DEFINITION 27. Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function. We define  $lss_q(f)$  to be the size of the smallest linear secret sharing scheme  $(\mathbb{F}_q, M, \psi)$  computing  $f$ . Similarly,  $\mu_q(f)$  denotes the minimum size when only considering multiplicative schemes and  $\mu_q^*(f)$  denotes the minimum size when only considering  $\mathcal{A}_f$ -strongly multiplicative schemes.

We define  $lss_q(f) = \infty$  if no linear secret sharing scheme computing  $f$  exists and similarly define  $\mu_q(f) = \infty$  (respectively  $\mu_q^*(f) = \infty$ ) if no ( $\mathcal{A}_f$ -strongly) multiplicative scheme computing  $f$  exists.

## 2.5. Secret Sharing

---

PROPOSITION 2. *Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function and  $\mathbb{F}_q$  any finite field. The following properties hold.*

- $lsss_q(f) < \infty$ .
- $lsss_q(f) \leq \mu_q(f) \leq \mu_q^*(f)$ .
- If  $f$  is  $Q^{(2)}$ , then  $\mu_q(f) < \infty$ .
- If  $f$  is  $Q^{(3)}$ , then  $\mu_q^*(f) < \infty$ .

Furthermore, we have the following relation between  $lsss_q(f)$  and  $\mu_q(f)$ .

THEOREM 14. *Let  $f$  be a monotone Boolean function that is  $Q^{(2)}$ . Then for any finite field  $\mathbb{F}_q$ ,*

$$\mu_q(f) \leq 2 \cdot lsss_q(f).$$

OPEN PROBLEM 1. *Do there exist constants  $q, c \geq 0$  such that*

$$\mu_q^*(f) \leq (\mu_q(f))^c$$

*for all  $Q^{(3)}$  monotone Boolean functions  $f$ ?*

### 2.5.4 Threshold Secret Sharing and Shamir's Secret Sharing Scheme

There is a special subclass of the perfect linear secret sharing schemes that is often considered, which is that of the  $t$ -threshold secret sharing schemes. These schemes have a particularly simple classification of subsets  $B \subset \mathcal{N}$ . In a  $t$ -threshold secret sharing scheme, all sets  $B \subset \mathcal{N}$  with  $|B| \leq t$  are rejected and all sets  $B \subset \mathcal{N}$  with  $|B| > t$  are accepted. Since all rejected sets have cardinality at most  $t$ , we say that the scheme is secure against a  $t$ -adversary. The most commonly used linear secret sharing scheme, which is due to Shamir [66], is such a  $t$ -threshold secret sharing scheme.

Shamir's secret sharing scheme [66] is based on the use of the Reed-Solomon codes described in Section 2.1 and can be described as follows. Let  $C \subset \mathbb{F}^{n+1}$  be an  $[n, t]$  Reed-Solomon code as described in Section 2.1. Then  $C$  is generated by the Vandermonde matrix

$$M = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^t \\ 1 & x_1 & x_1^2 & \cdots & x_1^t \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^t \end{pmatrix}.$$

Taking  $x_0 = 0$ , the matrix  $M$  now defines a linear secret sharing scheme via Definition 20, which has been known as Shamir's secret sharing scheme since its introduction in 1979.

It is easy to show that this secret sharing scheme is  $t$ -threshold. Since we already provide a proof for the more general class of secret sharing schemes based on MDS error correcting codes in Section 2.5.5, and Reed-Solomon codes are MDS, we omit the proof of this fact here.

As already mentioned in Section 2.3.5, the generalized Reed-Solomon codes can be seen as a subset of the algebraic-geometric codes, namely those algebraic-geometry codes that are defined over the projective line  $\mathbb{P}^1(\overline{\mathbb{F}}_q)$ . This suggests that it is possible to describe Shamir's secret sharing scheme in terms of algebraic-geometric coding techniques. We will show later on that this is indeed possible, and that this in fact leads to a strict generalization of the construction of Shamir's secret sharing scheme.

**THEOREM 15.** *Shamir's secret sharing scheme is multiplicative if  $n \geq 2t + 1$  and strongly multiplicative against a  $t$ -adversary if  $n \geq 3t + 1$ .*

*Proof.* We first note that any secret  $s_0$  with corresponding shares  $s_1, s_2, \dots, s_n$  in Shamir's secret sharing scheme correspond with a codeword  $(s_0, s_1, s_2, \dots, s_n) \in C$  and in fact all codewords in the Reed-Solomon code  $C$  can be retrieved in this manner. Furthermore, due to the construction of Reed-Solomon codes every codeword  $(s_0, s_1, s_2, \dots, s_n) \in C$  in turn corresponds exactly with a polynomial  $f \in \mathbb{F}_q[X]$  with  $\deg(f) \leq t$  such that  $f(x_i) = s_i$  for  $i = 0, 1, \dots, n$ .

We now look at two such codewords  $(s_0, s_1, s_2, \dots, s_n)$  and  $(t_0, t_1, t_2, \dots, t_n)$  in  $C$  with respective polynomials  $f$  and  $g$  and consider the product polynomial  $fg$ . Due to Lagrange's Interpolation Theorem it holds for a set  $B \subset \{1, 2, \dots, n\}$  that

$$s_0 \cdot t_0 = (fg)(x_0) = \sum_{i \in B} \lambda_i (fg)(x_i) = \sum_{i \in B} \lambda_i s_i t_i$$

for certain constants  $\{\lambda_i\}_{i \in B}$  in  $\mathbb{F}_q$ , provided that  $|B| \geq 2t + 1$ . If  $n \geq 2t + 1$  it now follows that the scheme is multiplicative. Furthermore, when all rejected sets have cardinality at most  $t$ , the complements of all rejected sets have cardinality at least  $2t + 1$  if  $n \geq 3t + 1$ . Therefore, Shamir's secret sharing scheme is strongly multiplicative with respect to a  $t$ -adversary if  $n \geq 3t + 1$ .  $\square$

## 2.5.5 Limitations of Treshold Schemes

Threshold secret sharing schemes are convenient to work with, but have some limitations that motivate us to consider more general constructions in Part III. The limitations are the following.

1. Each share is at least as large as the secret.

2. For any ideal  $t$ -threshold scheme, the finite field  $\mathbb{F}_q$  over which it is defined needs to be of size at least  $(n - 2)/2$  if  $1 \leq t \leq n - 3$ .

The first limitation holds simply due to the fact that the secret corresponds with exactly one row in the secret sharing matrix  $M$ , and shares correspond to at least one row in  $M$ . In fact, a similar statement holds more generally for any perfect secret sharing scheme without dummy indices as a consequence of Theorem 12 and can be found in Section 2.5.1.

The second restriction is more complex and is explained in detail in this section. Since it does not involve the special class of non-ideal threshold schemes, one could ask whether this class helps when trying to get around these restrictions. Unfortunately, while it is possible for non-ideal schemes to use fields of size smaller than  $n$ , the minimum average share size for these schemes is larger than  $\log_q(n/2)$ , which is even beyond that required for ideal threshold schemes.

### Ideal Threshold Schemes and Error Correcting Codes

It is well-known that there is a one-to-one correspondence between ideal  $t$ -threshold secret sharing schemes and maximum-distance-separable (MDS) error correcting codes. MDS codes are linear error correcting codes that meet the Singleton bound, i.e., the  $[n, k, d]$  codes for which  $k + d = n + 1$ . Below we first demonstrate the correspondence.

**LEMMA 3.** *If  $M$  is the  $(n + 1) \times e$  matrix corresponding to an ideal  $t$ -threshold secret sharing scheme, then*

1. *The rank of  $M$  is  $t + 1$ .*
2. *The rank of any  $(t + 1) \times e$ -submatrix of  $M$  is  $t + 1$ .*

*Proof.* First consider the  $(t + 1) \times e$ -submatrices  $N$  of  $M$  that do not include the first row of  $M$ . Since any such  $(t + 1) \times e$ -submatrix  $N$  of  $M$  corresponds with an accepted set, we must have that  $\vec{e}_1$  is in the image of  $N^T$ . If the rank of any such submatrix  $N$  is less than  $t + 1$ , this implies that there exists a  $t' \times e$  submatrix with  $t' < t + 1$  for which  $\vec{e}_1$  is in the image of  $N^T$ . This means that there exists a subset  $A$  with  $|A| = t' < t + 1$  that is accepted, which contradicts the fact that  $M$  gives a  $t$ -threshold scheme. Therefore, the rank of any such  $(t + 1) \times e$ -submatrix, and in particular the rank of  $M$ , is at least  $t + 1$ . This implies that the rank of *any*  $(t + 1) \times e$ -submatrix of  $M$  is  $t + 1$ , since any  $t \times e$  submatrix that doesn't include the first row has rank  $t$  and contains rows that are jointly linearly independent of  $\vec{e}_1$ .

We now demonstrate that the rank of  $M$  is at most  $t + 1$ . First assume that  $n > t + 1$ , since otherwise this trivially follows. We show that any set consisting of  $t + 2$  row vectors in  $M$ , say  $\{M_1, M_2, \dots, M_{t+2}\}$ , is linearly dependent, from which the claim follows. Since  $\vec{e}_1 \in \text{Im}(M_B^T)$  for any set  $B$  with  $|B| \geq t + 1$ , there exist

constants  $\lambda_i$  and  $\mu_i$  such that  $\vec{c}_1 = \sum_{i=1}^{t+1} \lambda_i M_i = \sum_{i=2}^{t+2} \mu_i M_i$ . It follows that  $M_{t+2} = (\lambda_1 M_1 + \sum_{i=2}^{t+1} (\lambda_i - \mu_i) M_i) / \mu_{t+2}$  and the  $t + 2$  rows are linearly dependent.  $\square$

**THEOREM 16.** *For any  $t, n \in \mathbb{Z}_{\geq 0}$  such that  $t < n$ , there is a one-to-one correspondence between the set of ideal  $t$ -threshold linear secret sharing schemes and the set of  $[n + 1, t + 1]$  MDS codes.*

*Proof.* ( $\Rightarrow$ ) Note that any  $t$ -threshold linear secret sharing scheme is generated by a  $(n + 1) \times e$  matrix  $M$ . Furthermore, the threshold property of the scheme implies that the rank of  $M$  and the rank of any  $(t + 1) \times (t + 1)$  submatrix of  $M$  is  $t + 1$ , so we can without loss of generality assume that  $e = t + 1$ . We now demonstrate that the code generated by this matrix is an  $[n + 1, t + 1]$  MDS code. It is then trivial to see that distinct secret sharing schemes give rise to distinct MDS codes via this construction.

Let  $\vec{b} \in \mathbb{F}^{t+1}$  be a vector generating a codeword  $\vec{c} = M\vec{b}$ . The first coefficient  $c_0$  of  $\vec{c}$  corresponds with a secret in the threshold secret sharing scheme generated by the matrix  $M$ . Consider a codeword  $\vec{c} = M\vec{b}$ . Since any  $t + 1$  positions in  $\vec{c}$  determine  $\vec{c}$ , any codeword containing at least  $t + 1$  zeroes is equal to the all-zero codeword. It follows that any non-zero codeword can contain at most  $t$  zeroes, which implies that the minimum distance  $d$  of the code is at least  $n + 1 - t$ . The dimension  $k$  is  $t + 1$ , so we obtain  $d \geq n + 1 - t = n + 1 - (k - 1) = n + 2 - k$ . Furthermore, by the singleton bound  $d + k \leq (n + 1) + 1$ , which implies that  $d + k = n + 2$  and the corresponding code is MDS.

( $\Leftarrow$ ) Let  $C$  be an  $[n + 1, t + 1, d]$  MDS code. The corresponding secret sharing scheme is the secret sharing scheme based on the generator matrix  $M$  of the code  $C$ . In Section 7.2 we prove that such a secret sharing scheme accepts all subsets  $A \subset \mathbb{P}$  with  $|A| \geq (n + 1) - d + 1$  and rejects all subsets  $A \subset \mathbb{P}$  with  $|A| \leq d^\perp - 2$ , where  $d^\perp$  is the minimum distance of the dual code  $C^\perp$ . Since the code is MDS we have that  $(n + 1) + 1 - d = k$ , so we all subset  $A$  of cardinality  $t + 1$  are accepted. Furthermore, we have that the  $[n + 1, n - t, d^\perp]$  dual code  $C^\perp$  of  $C$  is also MDS [60]. This implies that  $d^\perp - 2 = t$ , so the secret sharing scheme based on  $C$  is  $t$ -threshold.  $\square$

Due to Theorem 16 we know that any ideal threshold secret sharing scheme is equivalent to an MDS error correcting code. For any  $[n, k, d]$  MDS code  $C$  over a field  $\mathbb{F}_q$  it is known (see for instance [60], pages 94 – 95) that  $d = n - k + 1 \leq q$  if  $2 \leq k$ . Furthermore, since the dual  $[n, n - k, k + 1]$  code  $C^\perp$  is also MDS this implies that  $k + 1 \leq q$  if  $n - k \geq 2$ .

If we use these values with any MDS error correcting code and its dual code and apply Theorem 16, we obtain the following result.

**THEOREM 17.** *For any ideal  $t$ -threshold secret sharing*

$$q \geq \max\{n - t, t + 2\} \geq (n - 2)/2$$

## 2.5. Secret Sharing

---

when  $1 \leq t \leq n - 3$ .

It is worth noting that the Main Conjecture on MDS Codes even states that for any MDS code we necessarily have that  $q \geq n - 1$ .

### Bound on the Share Size of Threshold Schemes

We demonstrate now a result that shows that for threshold secret sharing schemes the average share size necessarily increases with the value of  $n$  when the threshold is *non-trivial* (i.e., not 0 or  $n - 1$ ). This claim is made more precise by a result of Cramer and Fehr [25], which is in turn based on a theorem by Karchmer and Wigderson [49].<sup>1</sup> Although the claim by Cramer and Fehr [25] is actually stated for binary fields, it is in fact fairly straightforward to see that it holds more generally for any arbitrary finite field. The part of the statement that is relevant here is the following.

**THEOREM 18.** ([25]) *For any  $t$ -threshold secret sharing scheme  $(\mathbb{F}_q, M, \psi)$  of size  $d$  with  $0 < t < n - 1$ ,*

$$d \geq n \cdot \log_q \frac{n + 3}{2}.$$

Informally, the theorem states that for any  $t$ -threshold secret sharing scheme, the average share size at least grows logarithmically in  $n$ . The bound is in fact similar to that arising from the correspondence to MDS codes, except that this statement also holds for non-ideal threshold schemes and does not directly imply bounds on the field size.

---

<sup>1</sup>Unlike the previous bounds obtained from the correspondence with MDS codes which have been shown by Chen, Cramer, Goldwasser, de Haan and Vaikuntanathan [17], the mention of these stronger bounds is novel in this work.



## Part II

# Perfectly Secure Message Transmission



## Chapter 3

# Background

### 3.1 Introduction

It often occurs that two parties, a sender and a receiver, want to transmit a message to one another while guaranteeing the authenticity of the message and safeguarding the contents of the message against any eavesdropping third party. Communication that meets these two requirements is called *secure message transmission*. For simplicity assume that the communication channel is perfect in the sense that transmission over the channel is error-free, but that there is some adversarial party that is able to eavesdrop on the communication over the channel. Perfect authenticity is trivially achieved here. Perfect privacy can be achieved in this setting by means of one-time-pad encryption, where the sender and receiver somehow at some point before the transmission agree on a secret key that consists of at least as many bits as the message and use this key to mask the message during the transmission. However, in order to achieve perfect privacy such a key can only be used once and therefore requires the storage of a lot of key-data in order to allow the transmission of a large number of messages.

If the power of the adversarial party is strengthened up to the point where it is able to modify data transmitted on the communication channel, things become much more problematic. In particular, all communication can be blocked and no communication can be guaranteed to arrive at its intended destination at all.

To get around these problems, Dolev, Dwork, Waarts and Yung [28] considered a multi-channel model. In this model, two parties are connected by  $n > 1$  communication channels and an adversarial party is able to eavesdrop and modify data, except that it is during any transmission restricted to operate on at most  $t$  channels, where  $t < n$ . In fact, this model can be seen as the natural abstraction of a typical communication network, where the channels depict all the vertex-disjoint communication

paths from the sending party to the intended receiving party.

This model has two important advantages. First of all, it is possible to prevent an adversary from totally blocking all communication, as at least one channel will always be out of reach. More importantly, when  $t$  is small enough compared to  $n$ , it is possible to achieve secure communication *without using any initial secret key*. This strongly separates this model from the traditional model with one communication channel where one always requires either a computational restriction on the adversary or a pre-distributed secret key.

Suppose all communication over the channels only occurs in one direction, i.e., all communication is from the sender to the receiver. When it is possible to achieve secure message transmission in this setting, which depends on the restrictions on the adversary, secure message transmission is usually achieved in a rather straightforward way using a combination of secret sharing and error correction.

Dolev et al. demonstrated that, if one allows interaction between the sender and the receiver, it is possible to achieve secure message transmission while tolerating a more powerful adversary than in the uni-directional case. In fact, when it is possible to achieve secure transmission in this interactive setting at all, one can also guarantee perfect privacy and a zero-error probability. Authentication is trivially achieved, as any message that is accepted by the receiving party automatically has to have been transmitted by the sending party. Due to these properties, these type of protocols are referred to as *perfectly secure message transmission* protocols.

Following the results by Dolev et al., there have been a series of articles attempting to improve the efficiency and communication complexity of perfectly secure message transmission protocols. This has culminated in efficient protocols with optimal communication complexity for any choices of  $n$  and  $t$  and any level of interaction between the sending party and the receiving party.

In this part we describe all these results on perfectly secure message transmission following the article by Dolev et al. in a modular way, emphasizing the improvements made between the different articles. It is our hope that anyone interested in the field will find all he needs to know about it here.

## Overview

In Section 3.2 we define the model for perfectly secure message transmission. In Section 3.3 we define perfectly secure message transmission protocols. In Section 3.4 we give a historical overview of the results that are presented here. In Chapter 4 we describe the results on single-phase perfectly secure message transmission. In Chapter 5 we describe the results on two-phase perfectly secure message transmission.

## 3.2 Model

Informally, secure message transmission entails the following. A sender tries to interactively transmit a message to an intended receiver using  $n$  disjoint communication channels, while an adversary can adaptively take control of up to  $t$  of the channels and can read and modify data transmitted on the channels under his control. The goal is to execute the transmission in such a way that, despite the efforts of the adversary, the message remains private with respect to the adversary and arrives correctly at the receiver.

More formally, we can model the setup as follows. There are three parties, a sender  $\mathcal{S}$ , a receiver  $\mathcal{R}$  and an adversary  $\mathcal{A}$  that can be seen as communicating deterministic processes. Both  $\mathcal{S}$  and  $\mathcal{R}$  are connected to  $\mathcal{A}$  by  $n$  ideal communication channels, labeled as  $\{s_i\}_{i=1}^n$  and  $\{r_i\}_{i=1}^n$  respectively, while no direct communication channels exist between  $\mathcal{S}$  and  $\mathcal{R}$ . We assume that all data transmitted arrives correctly and without delay, and that it can be determined when the transmission on any particular channel has completed. The setting is depicted in Figure 3.1.

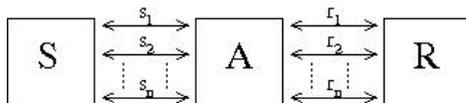


Figure 3.1: A schematic representation of the connectivity of  $\mathcal{S}$ ,  $\mathcal{A}$  and  $\mathcal{R}$ .

The execution of any message transmission protocol consists of sequential phases. At the start of the first phase  $\mathcal{S}$ ,  $\mathcal{R}$  and  $\mathcal{A}$  each have a private finite random string, while  $\mathcal{S}$  additionally has a private (random) message  $M$  from some public finite message set  $\mathcal{M}$ . Furthermore,  $\mathcal{A}$  initializes a private index set  $I_{\mathcal{A}} \subset \{1, 2, \dots, n\} := \mathcal{N}$  based on his random string. Once the message  $M$  and the random strings are fixed, the remainder of the protocol proceeds in a deterministic fashion, where additionally the round functions of  $\mathcal{S}$  and  $\mathcal{R}$  are public.

At the start of each phase  $\mathcal{S}$  and  $\mathcal{R}$  transmit some finite amount of data over the channels connecting the respective party to  $\mathcal{A}$ . This data is deterministically computed based on publicly and locally available information; the data received in previous phases, the phase number, the local private random string and additionally, for  $\mathcal{S}$ , the message  $M$ .

In every phase, after  $\mathcal{S}$  and  $\mathcal{R}$  are done transmitting data,  $\mathcal{A}$  proceeds as follows. For any channel  $s_i$  with  $i \notin I_{\mathcal{A}}$ ,  $\mathcal{A}$  forwards the data that is received on it unread and unmodified from  $\mathcal{S}$  to  $\mathcal{R}$  on channel  $r_i$  and vice versa. Additionally,  $\mathcal{A}$  reads all data on the channels  $\{s_i\}_{i \in I_{\mathcal{A}}}$  and  $\{r_i\}_{i \in I_{\mathcal{A}}}$  and then transmits some finite amount of data on the channels  $\{s_i\}_{i \in I_{\mathcal{A}}}$  and  $\{r_i\}_{i \in I_{\mathcal{A}}}$  that is computed based on all data read so far, the phase number and his local private random string. Finally, based on this

information,  $\mathcal{A}$  selects a number of indices in  $\mathcal{N}$  and adds these to the set  $I_{\mathcal{A}}$ .

At the end of the last phase,  $\mathcal{R}$  outputs a value  $M' \in \mathcal{M}$  based on all data received and the local random string.

### 3.3 Definitions

DEFINITION 28. *An adversary  $\mathcal{A}$  is a  $t$ -adversary if he can never control more than  $t$  communication channels, i.e.,  $|I_{\mathcal{A}}| \leq t$  at the end of any message transmission protocol  $\mathcal{A}$  participates in.*

DEFINITION 29. *The view  $V_{\mathcal{A}}$  of  $\mathcal{A}$  at the end of a protocol execution consists of its local random string and any data read by  $\mathcal{A}$  up to and during the last phase.*

DEFINITION 30. *A pair of algorithms for  $\mathcal{S}$  and  $\mathcal{R}$  is called a perfectly secure message transmission protocol if the following properties hold at the end of any protocol execution with a  $t$ -adversary  $\mathcal{A}$ :*

- (Correctness) *The message  $M'$  that  $\mathcal{R}$  outputs at the end of the protocol is equal to the message  $M$  held by  $\mathcal{S}$ , i.e.,*

$$P(M' = M) = 1.^1$$

- (Privacy) *For any possible view  $v_{\mathcal{A}}$  for  $\mathcal{A}$  at the end of a protocol execution and any  $M_1, M_2 \in \mathcal{M}$ ,*

$$P(M = M_1 | V_{\mathcal{A}} = v_{\mathcal{A}}) = P(M = M_2 | V_{\mathcal{A}} = v_{\mathcal{A}}).$$

*In other words, the view of  $\mathcal{A}$  is independent of the message  $M$ .*

*Here the probabilities are taken over the private random strings of  $\mathcal{S}$ ,  $\mathcal{R}$  and  $\mathcal{A}$ .*

Recall that any element of the finite message set  $\mathcal{M}$  can be uniquely described using  $\lceil \log_2 \mathcal{M} \rceil$  bits. Since we will assume in the sequel that elements of  $\mathcal{M}$  are selected uniformly at random, this is in fact optimal. The following definitions allow us to measure the efficiency of secure message transmission protocols.

DEFINITION 31. *The phase complexity  $p$  of a protocol is the number of subsequent communication phases between  $\mathcal{S}$  and  $\mathcal{R}$  required by the protocol. The communication complexity  $\mathcal{C}$  of a protocol is defined to be the number of bits that is transmitted by  $\mathcal{S}$  and  $\mathcal{R}$  during a worst-case protocol execution with an optimal adversary, overall, for the secure transmission of a message of length  $\ell$  bits. The communication overhead  $\Lambda$  of a protocol is defined to be  $\Lambda = \mathcal{C}/\ell$ , i.e., the number of bits to be communicated by  $\mathcal{S}$  and  $\mathcal{R}$  overall per bit of the original message.*

<sup>1</sup>If the condition is replaced by  $P(M' = M) > 1 - \epsilon$ , the protocol is said to be *probabilistically secure* with error probability  $\epsilon$ . We restrict the discussion to perfectly secure protocols here.

### 3.4. Historical Overview

---

Although we will not define this explicitly, one can also put a measure on the computation complexity of secure message transmission protocols. We are mainly interested in the distinction between computationally efficient and computationally inefficient protocols, where a protocol is considered efficient if it requires computation at most polynomial in  $n$ .

#### **Broadcast**

Suppose that  $n > 2t$  and that we require correctness, but not privacy during message transmission. Then there exists an extremely simple single-phase protocol against a  $t$ -adversary that achieves this level of security. The protocol works as follows.

1.  $\mathcal{S}$  sends a copy of the message  $M$  over every communication channel  $\{s_i\}_{i \in \mathcal{N}}$ .
2.  $\mathcal{R}$  reads the values received on the different channels and performs majority voting to determine  $M$ .

It is trivial to see that this protocol works, as  $\mathcal{A}$  can only send modified data to  $\mathcal{R}$  on a minority of the channels  $\{r_i\}_{i \in \mathcal{N}}$ . We call this method of reliably transmitting information *broadcast* in the sequel.

Obviously, if  $\mathcal{A}$  sends modified data to  $\mathcal{R}$  during this protocol then  $\mathcal{R}$  can immediately identify the corresponding indices in  $I_{\mathcal{A}}$ , while sending the modified data does not help  $\mathcal{A}$  eavesdrop on or disturb the protocol in any way. Since we are mainly interested in the worst-case adversaries when analyzing our protocols, we therefore assume without loss of generality in the sequel that  $\mathcal{A}$  forwards all data unmodified during a broadcast.

## 3.4 Historical Overview

In 1993, Dolev, Dwork, Waarts and Yung [29] started the line of research in perfectly secure message transmission, listing a number of important initial results. One observation was that single-phase PSMT protocols exist if and only if  $n \geq 3t + 1$ . This is consistent with comparable research in perfectly secure protocols against an active adversary, where it is commonly also required to have  $n \geq 3t + 1$ . Another example of such protocols are perfectly secure multi-party computation protocols, which are discussed in Part IV.

Dolev et al. furthermore discovered that as soon as you allow *interaction* in the message transmission protocol, i.e., multiple transmission phases where the parties can communicate feedback with regard to the data they received, it is possible to construct perfectly secure protocols under the weaker restriction  $n \geq 2t + 1$ . As to the required number of communication phases, they showed that for  $n \geq 2t + 1$

it is necessary and sufficient to use two phases. Finally, Dolev et al. showed that no (perfectly or probabilistically) secure message transmission protocols exist when  $n \leq 2t$ .

Following these results in [29], it remained an open question to determine the optimal achievable communication overhead for such protocols. We first consider one-phase protocols where  $n \geq 3t + 1$ . In 2004, Srinathan, Narayanan and Pandu Rangan [70] demonstrate that communication overhead  $n/(n - 3t)$  is achievable. It turns out  $n/(n - 3t)$  overhead is also necessary, as proven in 2007 by Fitzi, Franklin, Garay and Harsha Vardhan [33].

We now consider the setting where  $n \geq 2t + 1$  and two communication phases are allowed. An initial communication-inefficient protocol for this setting was given in [29]. The first communication-efficient protocol was later presented in 1996 by Sayeed and Abu-Amara [64], who achieve an  $\Omega(n^3/(n - 2t))$  communication overhead. In the 2004 paper [70], a lower bound of  $n/(n - 2t)$  is established for the communication overhead. We uncovered a fundamental flaw in another result of [70] (see [1]), but also discovered that some of the techniques from this paper allow to reduce the communication overhead of the Sayeed and Abu-Amara protocol to  $\Omega(n^2/(n - 2t))$ .

In 2006, Agarwal, Cramer and de Haan [1] achieve the optimal  $\Omega(n/(n - 2t))$  communication overhead. It is interesting to note that one cannot achieve a lower communication overhead through the use of more than two communication phases, as was later proved in 2007 by Srinathan, Prasad and Pandu Rangan [71]. This means that the research line concerning the communication overhead of perfectly secure message transmission protocols is now essentially closed.

As a final twist, Kurosawa and Suzuki [51] prove in 2008 that it is possible to achieve the optimal two-phase communication overhead for  $n \geq 2t + 1$  with a computationally efficient protocol, which was left as an open problem in [1]. Although not originally presented in this manner, it turns out that when one extracts the key idea from their article and swaps it into the framework of [1], one can achieve the same result. We present this approach in this work.

The results described up to now are tied to the worst-case two-phase setting where  $n = 2t + 1$ . If one moves an  $\epsilon$ -fraction away from these parameters, i.e., sets  $n = (2 + \epsilon)t$  with  $\epsilon > 0$ , one achieves a natural lower bound of  $\Omega(1)$  on the communication overhead. In fact, this overhead can be achieved using a straightforward modification of the protocol in [1], but this results in a computationally inefficient protocol. Fitzi et al. [33] present techniques that allow to achieve this lower bound with a computationally efficient protocol. Although the same effect can now also be achieved by a modification of the protocol in [51], the techniques in [33] are fundamentally different from those used in other message transmission protocols and interesting to discuss in their own right.

## Chapter 4

# Single-Phase Perfectly Secure Message Transmission

For completeness, we fully treat the known results on perfect, single-phase message transmission, where some of the results are also used in the multi-phase setting. In a sense, the single-phase setting can be seen as the simple case of the general problem, as the optimal communication overhead can already be achieved using a straightforward combination of secret sharing and error correction.

The following remarks are used to simplify the discussion in this chapter. When the communication consist of a single phase, it is easy to see that only the data transmitted by  $\mathcal{S}$  (and  $\mathcal{A}$ ) is relevant. Furthermore, we can consider the data transmitted by  $\mathcal{S}$  to be a codeword in a code  $C$  of length  $n$  and the data received by  $\mathcal{R}$  to be the same codeword in  $C$  after the introduction of up to  $t$  errors.

We start by demonstrating that  $n \geq 3t+1$  is a necessary condition for single-phase perfectly secure message transmission and proceed by showing that this condition is also sufficient. Finally, we argue the required and achievable communication overhead for such protocols.

We first fix some notation. Let  $\mathcal{M}$  be the message space from which the message  $M$  is drawn. Furthermore, let  $C$  be a code of length  $n$  over some finite field  $\mathbb{F}$  and  $\phi : C \rightarrow \mathcal{M}$  be a surjective map that maps a codeword  $\vec{c} \in C$  to a message  $M \in \mathcal{M}$ . We assume without loss of generality that the protocol consists of  $\mathcal{S}$  selecting a codeword  $\vec{c} \in \phi^{-1}(M)$  uniformly at random and transmitting it. At the end of the protocol,  $\mathcal{R}$  then receives a vector  $\vec{c}' \in \mathbb{F}^n$  for which  $d(\vec{c}, \vec{c}') \leq t$ .

For  $A \subset \{1, 2, \dots, n\}$ , we define  $C_A$  to be the restriction of  $C$  to the positions in  $A$ . When  $A$  is of the form  $\{i, i+1, \dots, j\}$ , we may also use notation  $C_{i,j}$  or  $C_i$  when  $i = j$ . We denote the subset of codewords corresponding to a message  $m \in \mathcal{M}$  by

$C^m$ , i.e.,  $C^m = \phi^{-1}(m)$ .

## 4.1 Necessity of $n \geq 3t + 1$

We can argue the necessity of  $n \geq 3t + 1$  as follows. First note that due to the privacy property, any data transmitted on any selection of up to  $t$  channels needs to be independent of the message  $M$ . The necessity of  $n \geq 3t + 1$  now follows from the following result from coding theory.

LEMMA 4. *For any one-phase reliable transmission, the data transmitted on any  $n - 2t$  channels necessarily determines the message  $M$ . [29]*

*Proof.* Note that when  $\mathcal{S}$  transmits a message  $M$  using some codeword  $\vec{c} \in C^M$ , the correctness property implies that  $M$  can be uniquely determined from any vector  $\vec{c}' \in \mathbb{F}^n$  for which  $d(\vec{c}, \vec{c}') \leq t$ .

Now assume that the lemma is false, i.e., there exists a codeword  $\vec{c} \in C$  such that some  $n - 2t$  of the positions do not fix any message in  $\mathcal{M}$ . Without loss of generality we can assume this involves the first  $n - 2t$  positions. Then for some distinct  $m, m' \in \mathcal{M}$  there exist codewords  $\vec{c}_m \in C^m$  and  $\vec{c}_{m'} \in C^{m'}$  such that  $(\vec{c}_m)_{1, n-a-b} = (\vec{c}_{m'})_{1, n-a-b}$  for some  $1 \leq a \leq t$  and  $0 \leq b \leq t$ .

Consider now the case where  $\mathcal{R}$  receives the vector  $\vec{e} \in \mathbb{F}^n$  that consists of the concatenation of  $(\vec{c}_m)_{1, n-a-b}$ ,  $(\vec{c}_m)_{n-a-b+1, n-b}$  and  $(\vec{c}_{m'})_{n-b+1, n}$ . Since  $a, b \leq t$ , neither the subvector  $(\vec{c}_m)_{n-a-b+1, n-b}$  nor the subvector  $(\vec{c}_{m'})_{n-b+1, n}$  determines the message. Now by application of the correctness property, the vector  $\vec{e}$  should be decoded as the message  $m$ , since  $d(\vec{c}_m, \vec{e}) \leq t$ . However, since  $d(\vec{c}_{m'}, \vec{e}) \leq t$ , the vector  $\vec{e}$  should also be decoded as the message  $m'$ . This contradicts the fact that the protocol satisfies the correctness property.  $\square$

COROLLARY 5. *For any one-phase perfectly secure message transmission protocol it holds that  $n \geq 3t + 1$ .*

*Proof.* Lemma 4 shows that in order to satisfy the correctness property, the data that  $\mathcal{S}$  transmits on any  $n - 2t$  channels should determine  $M$ . Furthermore, since any data transmitted on any selection of up to  $t$  channels needs to be independent of the message  $M$  to maintain privacy, we need to have that  $n - 2t > t$ . This implies that  $n \geq 3t + 1$ .  $\square$

## 4.2 Sufficiency of $n = 3t + 1$

Let  $\hat{C} \subset \mathbb{F}^{n+1}$  with  $|\mathbb{F}| > n$  be the  $[n+1, t+1]$  Reed-Solomon code with  $n+1 = 3t+2$ , where we number the positions in the codewords of  $\hat{C}$  using the numbers  $0, 1, \dots, n$ .

### 4.3. Communication Lower Bound for $n \geq 3t + 1$

---

Let  $C \subset \mathbb{F}^n$  be the  $[n, t + 1]$  Reed-Solomon code  $\hat{C}_N$  obtained after projecting on the last  $n$  positions of the codewords in  $\hat{C}$ . Now consider the following algorithm for  $\mathcal{S}$ :

1. Select a codeword  $\hat{c} \in \hat{C}$  uniformly at random under the restriction that the first coefficient is the message  $M \in \mathbb{F}$ .
2. Let  $\vec{c} \in C$  be the codeword that results after removing the first position in the codeword  $\hat{c}$ .
3. Transmit the value in the  $i^{\text{th}}$  position of  $\vec{c}$  over the channel  $s_i$  for  $i = 1, 2, \dots, n$ .

Since  $d_{\min}(C) = n - t = 2t + 1$ ,  $\mathcal{R}$  is able to correct errors introduced in any up to  $t$  positions in the codeword  $\vec{c}$ . Using any  $t + 1$  of the recovered positions,  $\mathcal{R}$  can now determine the codeword  $\hat{c}$  and thereby output  $M$ . On the other hand, for any up to  $t$  positions that  $\mathcal{A}$  reads and any value  $M' \in \mathbb{F}$  there exists a unique codeword in  $\hat{C}$  that contains the value  $M'$  in the first position and that matches the read positions. Therefore, given the values read by  $\mathcal{A}$ , any possible value in  $\mathbb{F}$  is equally likely to be the message that was transmitted by  $\mathcal{S}$ . Therefore, the protocol is both private and correct and is thus a perfectly secure message transmission protocol.

### 4.3 Communication Lower Bound for $n \geq 3t + 1$

It is easy to verify that the protocol listed above introduces a communication overhead of  $n$ , i.e., for every value in  $\mathbb{F}$  that  $\mathcal{S}$  transfers to  $\mathcal{R}$   $n$  values in  $\mathbb{F}$  need to be transmitted over the channels by  $\mathcal{S}$ . Fitzi, Franklin, Garay and Harsha Vardhan [33] show that this overhead is in fact optimal for  $n = 3t + 1$ .

**THEOREM 19.** *Any one-phase PSMT protocol for  $n > 3t$  channels requires communication overhead  $\geq n/(n - 3t)$ .*

*Proof.* Consider any one-phase PSMT protocol for  $n > 3t$  based on a code  $C \subset \mathbb{F}^n$ . Recall that it is required due to the privacy property that any  $t$  positions in the codeword are independent of the message  $M$ . In particular, this implies that for any two messages  $m_1, m_2 \in \mathcal{M}$  we have that  $C_{2t+1, 3t}^{m_1} = C_{2t+1, 3t}^{m_2}$ . Furthermore, perfect correctness implies that any  $n - 2t$  positions in the codeword should uniquely determine the message  $M$  and therefore  $C_{2t+1, n}^{m_1} \cap C_{2t+1, n}^{m_2} = \emptyset$  whenever  $m_1 \neq m_2$ . Note that in order for the latter condition to hold, there should exist at least one distinct configuration of values for the last  $n - 3t$  positions in the code for every message  $m \in \mathcal{M}$ . In other words,  $|C_{3t+1, n}| \geq |\mathcal{M}|$ . Since  $\prod_{i=3t+1}^n |C_i| \geq |C_{3t+1, n}|$  this implies that  $\prod_{i=3t+1}^n |C_i| \geq |\mathcal{M}|$ .

Due to symmetry, this inequality holds for any selection of  $d = n - 3t$  positions in  $C$ . If we now enumerate over all subsets of  $d$  consecutive integers in  $\{1, 2, \dots, n\}$ , allowing

'wrap-around' (including for instance the subset  $\{n-1, n, 1, 2, \dots, d-2\}$ ), and look at the corresponding positions in the code, we obtain  $n$  configurations that together count every position exactly  $d$  times. It follows that  $(\prod_{i=1}^n |C_i|)^d \geq |\mathcal{M}|^n$ . Since at least  $\log|C_i|$  bits are required to communicate the value in every  $i^{\text{th}}$  position, this implies that the overhead is at least  $(\sum_{i=1}^n \log|C_i|)/\log|\mathcal{M}| \geq n/d = n/(n-3t)$ .  $\square$

## Chapter 5

# Two-Phase Perfectly Secure Message Transmission

The first two-phase perfectly secure message transmission protocol was presented by Dolev, Dwork, Waarts and Yung [29] in 1993. However, the protocol introduced an exponential communication overhead. This made the protocol inefficient both in terms of communication overhead and computational complexity. In 1996, Sayeed and Abu-Amara [64] were the first to present an idea for a protocol that is efficient both in terms of communication overhead and computational complexity.<sup>1</sup> Their techniques allow to achieve a protocol with  $O(n^3)$  communication overhead.

Srinathan, Narayanan and Pandu Rangan [70] later, in 2004, present proof that every protocol with  $n = 2t + 1$  necessarily requires  $\Omega(n)$  communication overhead, and claim to present a protocol that achieves this optimal overhead. Although their protocol is incorrect (see [1]), it does involve a very useful technique that allows to reduce the communication overhead of perfectly secure message transmission protocols by replacing most broadcast occurrences with the application of an error correcting code. Applying this technique to the Sayeed-Abu-Amara protocol leads to a protocol with  $\Omega(n^2)$  communication overhead.

A few years later, in 2006, Agarwal, Cramer and de Haan [1] introduce a new idea that allows to achieve the optimal  $O(n)$  communication overhead. The basic idea is to ensure that all channels with data modifications can be detected and to organize the remainder of the protocol in such a way that  $\mathcal{R}$  can reconstruct all relevant parts of the initial data that  $\mathcal{S}$  received. This in turn allows to extract linear-size keys, as opposed to the constant-size keys that result from the Sayeed-Abu-Amara technique. However, the detection-technique presented in [1] requires the transmission of an

---

<sup>1</sup>It should be noted that their actual protocol contains an important error.

amount of data that is exponential in  $n$ , and therefore results in a computationally inefficient protocol.

As an intermediate result towards achieving both computational efficiency and optimal communication overhead, Fitzi, Franklin, Garay and Harsha Vardhan [33] in 2007 present a solution for the 'almost worst-case' parameters  $n = (2 + \epsilon)t$ . The protocol they present achieves constant communication overhead, which is trivially seen to be optimal.

Finally, in 2008 Kurosawa and Suzuki [51] present an improved detection-technique for modified data that only requires the transmission of a polynomial (in  $n$ ) amount of data. This leads to a computationally efficient protocol with the optimal  $\Omega(n)$  communication overhead.

Below we explain the ideas behind all of these protocols in a modular way using the framework first introduced in [1], which simplifies the task of identifying the conceptual improvements made between the various articles.

## 5.1 Sayeed and Abu-Amara's Protocol

The protocol by Sayeed and Abu-Amara [64] can be seen to consist of three parts. The first part establishes correlated data between  $\mathcal{S}$  and  $\mathcal{R}$  that is partially independent of any data read by  $\mathcal{A}$ . At the end of the second part  $\mathcal{S}$  and  $\mathcal{R}$  share identical data that is still partially independent of any data read by  $\mathcal{A}$ . The third part consists of a privacy amplification step that is then used to extract a mutual one-time-pad between  $\mathcal{S}$  and  $\mathcal{R}$  that is completely independent of any data read by  $\mathcal{A}$ . Such a one-time-pad can be used to encrypt the message and the resulting ciphertext can be transmitted to  $\mathcal{R}$  using broadcast (which involves a linear communication overhead).

In Section 5.1.1 we describe the first step of the protocol by means of the protocol  $\Pi_i$ , which is executed in parallel for every index  $i \in \mathcal{N}$ . Section 5.1.2 describes the information reconciliation step that enables  $\mathcal{S}$  and  $\mathcal{R}$  to agree on a partially private data vector and Section 5.1.3 describes the privacy amplification protocol that converts this vector into fully private data. Finally, Section 5.1.4 then shows how these steps combine into the full perfectly secure message transmission protocol.

### 5.1.1 Protocol $\Pi_i$

We now describe the two-phase subprotocol  $\Pi_i$  that has implicitly been used in the protocol of Sayeed and Abu-Amara [64]. Conceptually, during this protocol  $\mathcal{R}$  transmits a value  $s \in \mathbb{F}$  to  $\mathcal{A}$  over channel  $r_i$  and obtains feedback in the following phase about the value that is subsequently transmitted by  $\mathcal{A}$  on channel  $s_i$ . It has the following properties:

- If  $\mathcal{A}$  does not forward all data  $\mathcal{R}$  transmitted on channel  $r_i$  unaltered,  $\mathcal{R}$  can identify at the end of the second phase that  $i \in I_{\mathcal{A}}$ .
- If channel  $i$  is not contained in the set  $I_{\mathcal{A}}$  at the start of the first phase
  - $\mathcal{R}$  can identify that  $\mathcal{A}$  correctly forwarded  $s$ .
  - The total data that  $\mathcal{A}$  reads during the protocol is uncorrelated with  $s$ .

The details of the protocol are as follows. Assume that  $\mathbb{F} = \mathbb{F}_q$  with  $|\mathbb{F}_q| > n$  and let  $C \subset \mathbb{F}_{q^{t+1}}^{n+1}$  be an  $i$ -concentrated  $[n+1, t+1]$  Reed-Solomon code over  $\mathbb{F}_q$ . First,  $\mathcal{R}$  selects a codeword of the form  $(s, c_1, \dots, c_{i-1}, \alpha, c_{i+1}, \dots, c_n) \in C$  uniformly at random.  $\mathcal{R}$  then transmits  $\alpha$  over channel  $r_i$  and  $c_j$  on every remaining channel  $r_j$ . We denote the value transmitted by  $\mathcal{A}$  on channel  $s_i$  by  $\alpha'$  and the values transmitted on the remaining channels  $s_j$  by  $c'_j$ . This completes the first phase.

The second phase proceeds as follows. If  $\mathcal{S}$  receives incorrectly formed data on any of the channels,  $\mathcal{S}$  broadcasts a notification for these channels. Otherwise, for every pair of values such that  $f_{\alpha'}(x_j) \neq c'_j$ ,  $\mathcal{S}$  broadcasts  $j$ ,  $f_{\alpha'}(x_j)$  and  $c'_j$ .<sup>2</sup> Finally,  $\mathcal{R}$  verifies for all received values whether  $f_{\alpha'}(x_j) = c_j$  and identifies that  $i \in I_{\mathcal{A}}$  if this is not the case or if  $\mathcal{S}$  sent a notification for channel  $s_i$ . Otherwise,  $\mathcal{R}$  concludes that  $\mathcal{A}$  forwarded the value  $\alpha$  correctly, which implies that  $\mathcal{S}$  can compute  $s = f_{\alpha}(x_0)$ .

First note that at least  $t+1$  channels transmit correct values and any  $t+1$  correct values determine the codeword. Therefore it follows that whenever  $\alpha \neq \alpha'$  there is at least one correctly forwarded value  $c_j$  such that  $f_{\alpha'}(x_j) \neq c_j$ , which  $\mathcal{R}$  can identify. In particular, when none of the values that  $\mathcal{R}$  receives leads to such a contradiction this implies that  $\alpha = \alpha'$ . The remaining privacy property follows from a straightforward application of the properties of the code  $C$ .

### 5.1.2 Information Reconciliation

In this section we describe an information reconciliation technique that is based on an idea by Sayeed and Abu-Amara [64]. We assume that  $\mathcal{S}$  has a vector consisting of  $n = 2t + 1$  uniformly random values and that at least  $t + 1$  of these values are known by  $\mathcal{R}$ . Furthermore, suppose that  $\mathcal{A}$  read at most  $t$  of these values and that the additional data read by  $\mathcal{A}$  is not correlated with the remaining  $n - t$  values. The goal is to have  $\mathcal{S}$  transmit enough information to allow  $\mathcal{R}$  to recover the random vector while guaranteeing that the data read by  $\mathcal{A}$  remains completely uncorrelated with at least one of coefficients in the vector.

Concretely, let  $\mathbb{F}_q$  be a finite field with  $|\mathbb{F}_q| \geq n + t$  and assume that  $\mathcal{S}$  has a uniformly random vector  $\vec{v} = (v_1, v_2, \dots, v_n) \in \mathbb{F}_q^n$ . We now consider the (unique)

---

<sup>2</sup>It is easy to see that it is not really necessary here to transmit the values  $c'_j$ , but it is added here to simplify the presentation later on.

codeword  $\vec{c}$  in the  $[n + t, n]$  Reed-Solomon code  $C \subset \mathbb{F}_q^{n+t}$  for which the first  $n$  coefficients equal those in  $\vec{v}$ . Let  $\mathcal{S}$  broadcast the last  $t$  coefficients of  $\vec{c}$  to  $\mathcal{R}$ .

$\mathcal{R}$  now knows at least  $n$  coefficients of  $\vec{c}$ , which fixes the codeword  $\vec{c}$  and in particular the first  $n$  coefficients corresponding with  $\vec{v}$ . Since  $\mathcal{A}$  obtains the value for at most  $n - 1$  different positions of  $\vec{c}$ , it follows from the properties listed for Reed-Solomon codes in Section 2.1 that any coefficient that  $\mathcal{A}$  does not obtain is uncorrelated with the  $n - 1$  coefficients that  $\mathcal{A}$  does obtain. Therefore, the requirements are met.

### 5.1.3 Privacy Amplification

We now describe a well-known technique for perfect privacy amplification [6, 5], that is very well-suited for use in perfectly secure message transmission protocols. Suppose  $\mathcal{S}$  and  $\mathcal{R}$  share  $b$  uniformly random elements in  $\mathbb{F}_q$  and that it is promised that  $a < b$  of these elements are completely uncorrelated with the data that  $\mathcal{A}$  has access to. Then there is a simple technique that allows  $\mathcal{S}$  and  $\mathcal{R}$  to non-interactively generate  $a$  of such random elements.

Assume that  $|\mathbb{F}_q| > a + b$  and let  $C \subset \mathbb{F}_q^{a+b}$  be a  $[a + b, b]$  Reed-Solomon code. Then we can view the  $b$  shared random elements as the first  $b$  coefficients of a (hereby uniquely determined) codeword  $\vec{c} \in C$ . By the properties of the Reed-Solomon code, the data accessible by  $\mathcal{A}$  is completely uncorrelated with the last  $a$  coefficients in  $\vec{c}$ . These coefficients can therefore be taken as the outcome of the privacy amplification.

### 5.1.4 The Protocol

The two-phase protocol due to Sayeed and Abu-Amara [64] is easily explained in terms of the techniques described in the previous sections. Initially,  $\Pi_i$  is executed in parallel for every channel  $i \in \mathcal{N}$ . This results in  $n$  random values  $\{v_1, v_2, \dots, v_n\}$  that are received by  $\mathcal{S}$ , of which at least  $t+1$  are equal to values that were originally transmitted by  $\mathcal{R}$ . Furthermore,  $\mathcal{R}$  finds out in the second phase which values were correctly received. Also,  $\mathcal{A}$  knows at most  $t$  of the values received by  $\mathcal{S}$ , which correspond to the indices in  $I_{\mathcal{A}}$ .  $\mathcal{S}$  and  $\mathcal{R}$  can now apply the information reconciliation technique from Section 5.1.2 and the privacy amplification technique from Section 5.1.3 to obtain a completely secret element  $v \in \mathbb{F}_q$ , which can then be used as a one-time pad.

## 5.2 A Lower Bound for Two-Phase PSMT

In this section we show the lower bound result on the communication overhead of perfectly secure message transmission protocols due to Srinathan, Narayanan and Pandu Rangan [70]. They prove the following theorem.

### 5.3. Improved Reliable Transmission

---

**THEOREM 20.** *Any two-phase perfectly secure message transmission protocol for  $n > 2t$  channels requires communication overhead  $\geq n/(n - 2t)$ .*

*Proof.* We mainly consider a weaker type of message transmission protocol, where we require correctness but not privacy, i.e., the only condition is that  $\mathcal{R}$  always correctly outputs the message  $M$ . The result then in particular holds for any full-fledged perfectly secure message transmission protocol. Furthermore, since correctness should hold for any random inputs for  $\mathcal{S}$ ,  $\mathcal{R}$  and  $\mathcal{A}$ , we can without loss of generality assume that the random input strings for  $\mathcal{S}$  and  $\mathcal{R}$  (that are independent of the message  $M$ ) are fixed and public. This makes the protocol deterministic.

We can now make some additional simplifying assumptions. Note that any data transmitted by  $\mathcal{S}$  in either the first or the second phase is independent of the data transmitted by  $\mathcal{R}$  in the second phase, so we can assume that  $\mathcal{R}$  only transmits data in the first phase. However, since all inputs for  $\mathcal{R}$  are fixed and public,  $\mathcal{S}$  can simulate the transmissions by  $\mathcal{R}$  locally. This implies that  $\mathcal{R}$  is not required to transmit any data at all. In particular  $\mathcal{S}$  only requires one phase to transmit his data. The bound now follows directly using the techniques in the proof of Theorem 19.  $\square$

## 5.3 Improved Reliable Transmission

The (potentially) most communication-intensive transmission in the protocol from Section 5.1 is the reliable transmission of the collisions during the second phase, i.e., the broadcasts involving the values  $f_{\alpha'}(x_j)$  and  $c'_j$  for which  $f_{\alpha'}(x_j) \neq c'_j$ . In this section we demonstrate a technique introduced by Srinathan, Narayanan and Pandu Rangan [70] that reduces the communication cost by replacing the broadcasts with a combination of broadcast and error correcting.

Let the set  $X := \{(i, j, a_j, b_j)\}$  consist of the collision values  $j$ ,  $a_j := f_{\alpha'}(x_j)$  and  $b_j := c'_j$  that need to be broadcast during the execution of the protocol  $\Pi_i$  for  $i = 1, 2, \dots, n$  as described in Section 5.1.1. Now define the undirected graph  $\mathcal{G} = (\mathcal{N}, \mathcal{E})$  by

$$(i, j) \in \mathcal{E} \Leftrightarrow (i, j, a_j, b_j) \in X \vee (j, i, a_i, b_i) \in X$$

and let  $M$  be the size of a maximum matching on  $\mathcal{G}$ . Since at least one of the two values  $a_j, b_j$  in every tuple is incorrect, every edge in this graph involves at least one channel on which data has been modified by  $\mathcal{A}$ . Therefore there are at least  $M$  channels that have transmitted modified values.

Furthermore, it can be seen that if the tuples corresponding to these  $M$  edges are broadcast by  $\mathcal{S}$ ,  $\mathcal{R}$  can determine at least  $M$  such channels. This implies that  $\mathcal{R}$  will be able to discard the values received on at least  $M$  channels during the remainder of the reliable transmission. Therefore, we can use an error correcting code with codewords of length  $n$  that can handle  $M$  erasures and  $t - M$  errors for the transmission. In

other words,  $n - M - 2(t - M) = M + 1$  values can now be transmitted using a single codeword of length  $n$ .

Since every edge in the graph involves at least one channel that is in the maximum matching, there can be at most  $2Mn$  edges in the graph. In particular, this implies that every set  $X$  can contain at most  $4Mn$  vectors. Using an error correcting code, every set  $X$  can thus be transmitted by sending  $O(n^2)$  values over the channels. This gives a linear improvement over the previous approach, where all of these values needed to be broadcast.

## 5.4 Communication-Optimal PSMT for $n = 2t + 1$

In this section we describe the details of the protocol due to Agarwal, Cramer and de Haan [1], which is the first known perfectly secure message transmission protocol that achieves the optimal linear communication overhead. Conceptually, the authors replace the subprotocol  $\Pi_i$  that establishes correlation between  $\mathcal{S}$  and  $\mathcal{R}$  from Section 5.1.1 and the protocol for information reconciliation as described in Section 5.1.2 by a new subprotocol  $\hat{\Pi}_i$ . This new subprotocol  $\hat{\Pi}_i$  has the following properties:

- At the end of the protocol,  $\mathcal{S}$  and  $\mathcal{R}$  both obtain a uniformly random vector  $\mathcal{Z}_i = (z_1, z_2, \dots, z_d) \in \mathbb{F}^d$ . However, they do not necessarily control which vector this is.
- If  $i \notin I_{\mathcal{A}}$ , the data read by  $\mathcal{A}$  is uncorrelated with the vector  $\mathcal{Z}_i$ .

Here  $d \in \mathbb{Z}_{>0}$  is some constant value that can be selected before the start of the protocol.

We compare the protocol  $\hat{\Pi}_i$  with the protocol  $\Pi_i$ . As shown in the protocol due to Sayeed and Abu-Amara (Section 5.1), after  $\Pi_i$  has been invoked once for every channel, up to  $t$  of the values that were actually received by  $\mathcal{S}$  may be unknown to  $\mathcal{R}$ . Therefore, almost all privacy had to be sacrificed during information reconciliation. However, when the protocol  $\hat{\Pi}_i$  finishes, information reconciliation has already occurred.

Furthermore, we show that by choosing a message  $m$  of sufficiently large size, the relative amount of privacy that has to be given up during the information reconciliation in the new protocol can be made arbitrarily small, whereas in the protocol due to Sayeed and Abu-Amara this amount is always proportional to the message size.

### 5.4.1 Sketch of the Techniques Used

During the protocol  $\Pi_i$  it is possible that data is replaced by  $\mathcal{A}$  on certain channels in the first phase, while these channels cannot be detected due to the fact that  $\mathcal{R}$  does

not receive sufficient feedback concerning these channels. The first part of protocol  $\hat{\Pi}_i$  introduces a technique that allows *all* channels with data modifications to be detected, even if only a single value has been altered on the channel. This then replaces the part of the original protocol that establishes the correlation between  $\mathcal{S}$  and  $\mathcal{R}$ .

The second part of the protocol  $\hat{\Pi}_i$  then consists of a new information reconciliation subprotocol. During this part of the protocol  $\mathcal{S}$  again sends some additional data, which turns out to be similar to the conflict information that is broadcast during the protocol  $\Pi_i$ . Due to this part of the protocol,  $\mathcal{R}$  can in the end reconstruct *all* values that  $\mathcal{S}$  received after the first phase, instead of just a restricted part of the data as was the case during the protocol  $\Pi_i$ .

The key to detecting all corrupted channels is the fact that there are always  $t + 1$  channels on which the data is forwarded unseen and unmodified by  $\mathcal{A}$ . Due to the use of (extended) Reed-Solomon codes, the values corresponding to these channels completely determine the original codeword and therefore any combination of the  $t + 1$  correct values together with one altered value will not correspond with any correct codeword in the code. Such inconsistencies allow  $\mathcal{R}$  to detect modifications on channels, since  $\mathcal{S}$  can send such altered values to  $\mathcal{R}$  and  $\mathcal{R}$  can verify that a value has actually been altered. However, since the set  $I_{\mathcal{A}}$  is not known and therefore also not the indices for the  $t + 1$  unmodified channels, this procedure needs to be repeated for all subsets consisting of  $t + 1$  channels to make sure that the proper subset of  $t + 1$  unmodified channels has been attempted.

### 5.4.2 Details of Protocol $\hat{\Pi}_i$

We now describe the protocol  $\hat{\Pi}_i$ , starting with the part that establishes the correlation. Let  $\mathbb{F}_q$  be such that  $|\mathbb{F}_q| > n + t$  and recall that  $\mathcal{N} = \{1, 2, \dots, n\}$ . The first phase of the protocol  $\hat{\Pi}_i$  is just as in the protocol  $\Pi_i$ , except that instead of one codeword,  $m$  codewords are initially selected and transmitted. Let  $C \subset \mathbb{F}_{q^{t+1}}$  be an  $i$ -concentrated  $[n, t + 1]$  Reed-Solomon code. As before, we denote the subspace of a code  $C$  that results from restricting to the positions in a set  $A \subset \{1, 2, \dots, n\}$  by  $C_A$  and similarly denote the restriction of a vector  $\vec{c} \in \mathbb{F}_q^n$  to  $A$  by  $(\vec{c})_A$  or  $(\vec{c})_j$  when  $A = \{j\}$ . Furthermore, let  $D \subset \mathbb{F}_q^n$  be the related Reed-Solomon code where in every codeword  $\vec{c} \in C$  the  $i^{\text{th}}$  coefficient  $\alpha$  is replaced by the value  $f_\alpha(x_i)$ .

1) *Phase 1:* In the first phase,  $\mathcal{R}$  initially selects  $m$  random codewords

$$\vec{c}_1, \vec{c}_2, \dots, \vec{c}_m \in C.$$

$\mathcal{R}$  then proceeds by transmitting  $\vec{c}_1, \vec{c}_2, \dots, \vec{c}_m$  as usual; by transmitting the respective values  $\{(\vec{c}_k)_j\}_{k=1}^m$  over channel  $r_j$  for  $j = 1, 2, \dots, n$ .

2) *Round 2:* Assume that  $\mathcal{S}$  receives the vectors  $\vec{c}'_1, \vec{c}'_2, \dots, \vec{c}'_m \in \mathbb{F}_{q^{t+1}}^n$ . To simplify the discussion we first replace these vectors by the corresponding vectors

$\vec{d}'_1, \vec{d}'_2, \dots, \vec{d}'_m \in \mathbb{F}_q^n$  where the heavy  $i^{\text{th}}$  coefficients  $(\vec{c}'_j)_i$  are replaced by the values  $f_{(\vec{c}'_j)_i}(x_i)$  in the appropriate vectors.

We now perform the verification as described in Section 5.4.1, where for every position  $j$  and every combination of  $t + 1$  positions  $j_1, j_2, \dots, j_{t+1}$  not including position  $j$  we try to find an index  $\ell$  for which the partial codeword  $(\vec{d}'_\ell)_A$  with  $A = \{j_1, j_2, \dots, j_{t+1}, j\}$  does not occur in the corresponding subcode  $D_A$ . For every such selection of channels, if such an index  $\ell$  exists  $\mathcal{S}$  broadcasts the value  $(\vec{d}'_\ell)_j$  and its index  $\ell$  to  $\mathcal{R}$ , who can then verify whether this value matches the transmitted data. In Lemma 5, we show that this approach allows  $\mathcal{R}$  to identify all channels  $j$  on which data has been replaced by  $\mathcal{A}$ .

It is clear that in the above procedure many values are broadcast that were not correlated with any data that was initially read by  $\mathcal{A}$ . In order to remove the newly introduced correlation, all codewords corresponding to the broadcast values are discarded. Of the remaining codewords, the values at the  $i^{\text{th}}$  positions are stored. Due to the similarities with the protocol  $\Pi_i$ , it now follows that these values are known to  $\mathcal{A}$  only if channel  $i$  has been read, whereas the data read by  $\mathcal{A}$  remains uncorrelated with these values otherwise.

Concretely, for the values  $j = 1, 2, \dots, n$ , let the set  $\mathcal{Q}_j = \{V_{j1}, V_{j2}, \dots, V_{jw}\}$  consist of all combinations of  $t + 1$  positions that do not include position  $j$ , i.e.,  $\mathcal{Q}_j = \{V \subset \mathcal{N} \setminus \{j\} : |V| = t + 1\}$ . Then for all members of the set  $\mathcal{Q}_j$  ( $1 \leq j \leq n$ ) the corresponding positions in a codeword determine the full codeword and every set  $\mathcal{Q}_j$  has the same number of elements (namely  $w = \binom{n-1}{t+1}$  elements).

The following protocol now specifies the verification step that is performed after the first phase.

**PROTOCOL A: CLASSIFY CHANNELS**

1. Let  $j \in \mathcal{N}$ ,  $k \in \{1, \dots, w\}$  and define  $W_{jk} := V_{jk} \cup \{j\}$ . Then either  $(\vec{d}'_\ell)_{W_{jk}} \in D_{W_{jk}}$  for every  $\ell \in \{1, \dots, m\}$ , or there is a smallest integer  $\ell_{jk}$  such that  $(\vec{d}'_{\ell_{jk}})_{W_{jk}} \notin D_{W_{jk}}$ .

Taking  $\ell_{jk} = 0$  when  $(\vec{d}'_\ell)_{W_{jk}} \in D_{W_{jk}}$  for every  $\ell$ , we let  $L_j = (\ell_{j1}, \dots, \ell_{jw})$  be the vector containing all such smallest indices,  $I_j = \{\ell_{j1}, \dots, \ell_{jw}\} \setminus \{0\}$  be the corresponding set of indices and define

$$E_j := ((\vec{d}'_{\ell_{jm}})_j)_{m \in \{1, \dots, w\}: \ell_{jm} \neq 0}$$

2. For  $j = 1, \dots, n$ ,  $\mathcal{S}$  broadcasts  $L_j$  and  $E_j$ . Furthermore,  $\mathcal{S}$  defines

$$\mathcal{Z}_i := ((\vec{d}'_\ell)_i)_{\ell \in \{1, \dots, m\} \setminus (\cup_{j=1}^n I_j)}.$$

We now specify the second part of the protocol  $\hat{\Pi}_i$ , consisting of Protocol B and Protocol C below, that performs the information reconciliation between the data of

$\mathcal{S}$  and  $\mathcal{R}$ . Note that during Protocol B almost the same ‘conflict information’ is transmitted as in the second phase of  $\Pi_i$ , where here it is crucial that whenever  $f_{(\vec{c}_\ell)_i}(x_j) \neq (\vec{c}_\ell)_j$  both conflicting values are returned instead of just one of these values.

However, this information is used in a completely different way. Whereas in previous protocols this information was required to discover channels on which data was modified, that functionality is now completely superfluous due to the previous part of the protocol. Instead, the information transmitted during Protocol B is exactly sufficient to allow for complete information reconciliation by  $\mathcal{R}$ , in the sense that it helps  $\mathcal{R}$  to completely determine what  $\mathcal{S}$  received in the first phase.

PROTOCOL B: GATHER RECONCILIATION INFORMATION

1. Define

$$\mathcal{C}_i := \{(\ell, j, (f_{(\vec{c}_\ell)_i}(x_j), (\vec{c}_\ell)_j)) : f_{(\vec{c}_\ell)_i}(x_j) \neq (\vec{c}_\ell)_j, j \in \mathcal{N} \setminus \{i\}, \ell \in \{1, \dots, m\}\}.$$

2.  $\mathcal{S}$  broadcasts  $\mathcal{C}_i$ .

After  $\mathcal{S}$  has finished transmitting,  $\mathcal{R}$  can now execute the following protocol to reconstruct  $\mathcal{Z}_i$ . Note that nothing needs to be transmitted anymore at this point.

According to Lemma 5, a channel  $s_j$  transmitted modified values if and only if there exists an entry in  $E_j$  that is inconsistent with the data transmitted by  $\mathcal{R}$ . This allows  $\mathcal{R}$  to completely split up the set of channels  $\{s_1, s_2, \dots, s_n\}$  in a set  $U_c$  of channels that transmitted modified values and a set  $U_u$  of unmodified channels. Together with the transmitted reconciliation data this allows to recover the data received by  $\mathcal{S}$ . Below we only demonstrate how  $\mathcal{R}$  can recover the vector  $\mathcal{Z}_i$ , but it is straightforward to see that the same technique can be extended to recover the remaining data received by  $\mathcal{S}$ . Since this extra data is not required for the protocol, this extension is omitted here.

PROTOCOL C: RECONCILE

1. First assume that  $i \in U_u$ . Then

$$\begin{aligned} \mathcal{Z}_i &= ((\vec{d}_\ell)_i)_{\ell \in \{1, \dots, m\} \setminus (\cup_{j=1}^n I_j)} \\ &= (f_{(\vec{c}_\ell)_i}(x_i))_{\ell \in \{1, \dots, m\} \setminus (\cup_{j=1}^n I_j)}, \end{aligned}$$

which is a vector known to  $\mathcal{R}$ .

2. Now assume that  $i \in U_c$ . Fix  $\ell \in \{1, \dots, m\}$ , let  $H \subset U_u$  be a set of  $t + 1$  unmodified channels and take  $j \in H$ . Then either  $(\vec{c}_\ell)_j = f_{(\vec{c}_\ell)_i}(x_j)$

or  $(\ell, j, f_{(\vec{c}_\ell)_i}(x_j), (\vec{c}_\ell)_j) \in \mathcal{C}_i$ . In the first case,  $(\vec{c}_\ell)_j$  (which is then equal to  $f_{(\vec{c}_\ell)_i}(x_j)$ , which is known to  $\mathcal{R}$ ) gives the value at the  $j^{\text{th}}$  position of the codeword  $(\vec{c}_\ell)_H \in C_H$  and in the second case  $f_{(\vec{c}_\ell)_j}(x_j)$  gives the value at the  $j^{\text{th}}$  position of the codeword  $(\vec{c}_\ell)_H \in C_H$ .

Since the values at any  $t + 1$  positions of a codeword in  $(\vec{c})_{H \cup \{i\}} \in C_{H \cup \{i\}}$  determine  $(\vec{c})_{H \cup \{i\}}$ , they in particular fix the value  $f_{(\vec{c})_i}(x_i)$ . Therefore, it follows that  $\mathcal{R}$  can obtain  $\mathcal{Z}_i$  in this case as well.

It follows that  $\mathcal{R}$  and  $\mathcal{S}$  both obtain the same vector  $\mathcal{Z}_i$  at the end of the protocol. If  $\mathcal{Z}_i$  is not empty, the values in the vector are either completely known to  $\mathcal{A}$  or he has no information about these values, depending only on whether  $i \in I_{\mathcal{A}}$  or not. This completes the description of protocol  $\hat{\Pi}_i$ .

### 5.4.3 The Protocol

Assume that we execute the protocol  $\hat{\Pi}_i$  in parallel for all  $n$  channels. Without loss of generality we may assume that all vectors  $\mathcal{Z}_i$  have the same length, since otherwise  $\mathcal{S}$  and  $\mathcal{R}$  can just remove entries according to some predetermined method. Also, it should be clear that for any  $i$  that the protocol is executed for, the set  $\bigcup_{j=1}^n I_j$  can contain at most  $nw = n \binom{n-1}{t+1}$  indices. Therefore, by choosing  $m$  large enough, the length of the vectors  $\mathcal{Z}_i$  can in fact be fixed to any nonzero value, so we can assume that the vectors  $\mathcal{Z}_i$  have nonzero length.

At most  $t$  of the vectors  $\mathcal{Z}_i$  have been read by  $\mathcal{A}$  at the end of the  $n$  parallel executions of  $\hat{\Pi}_i$ , whereas the remaining vectors are uncorrelated with the information read by  $\mathcal{A}$ . Therefore, applying a parallel version of the privacy amplification technique from 5.1.3 on these  $n$  vectors gives  $t + 1$  completely secret vectors. The values in these vectors can then be used in the second phase by  $\mathcal{S}$  to one-time-pad encrypt message elements from  $\mathbb{F}_q$ .

### 5.4.4 Proofs

We now provide the results that support the claims. The following lemma shows that  $\mathcal{R}$  can determine for a channel  $s_j$  whether any incorrect data has been transmitted on it in the first phase of  $\hat{\Pi}_i$  by comparing the received values in the set  $E_j$  with the original values that were transmitted on channel  $r_j$  in the first phase.

As described in the beginning of Section 5.4.2, let  $D \subset \mathbb{F}_q^n$  be the Reed-Solomon code obtained from  $C$  by replacing the  $i^{\text{th}}$  coefficient  $\alpha$  with  $f_\alpha(x_i)$ . Similarly, we define vectors  $\vec{d}_j$  related in this way to the transmitted vectors  $\vec{c}_j$  and vectors  $\vec{d}'_j$  related to the received vectors  $\vec{c}'_j$ .

LEMMA 5. Fix any  $j \in \mathcal{N}$ . Then  $(\vec{d}'_\ell)_j \neq (\vec{d}_\ell)_j$  for some  $\ell \in \{1, \dots, m\}$  if and only if there is a value  $\ell' \in \{1, \dots, m\}$  such that  $(\vec{d}'_{\ell'})_j \neq (\vec{d}_{\ell'})_j$  and  $(\vec{d}_{\ell'})_j$  is an entry of  $E_j$ .

*Proof.* ( $\Leftarrow$ ) Trivial.

( $\Rightarrow$ ) If  $(\vec{d}'_j)_\ell \neq (\vec{d}_j)_\ell$  for some  $\ell \in \{1, \dots, m\}$ , then there is a set  $V_{jk}$  of  $t+1$  indices that are not in  $I_{\mathcal{A}} \cup \{j\}$ . In particular,  $(\vec{d}'_j)_{W_{jk\ell}} \notin D_{W_{jk\ell}}$ , since the corresponding  $t+1$  correctly received coefficients in the received vector only occur in a codeword in  $D$  where the  $j^{\text{th}}$  coefficient is  $(\vec{d}_j)_\ell$ , which is different from  $(\vec{d}'_j)_\ell$ .

Now let  $\ell'$  be the smallest value for which  $(\vec{d}'_j)_{W_{jk\ell'}} \notin D_{W_{jk\ell'}}$ . Since the indices in  $V_{jk}$  are not in  $I_{\mathcal{A}}$ , the values at the positions in the set  $V_{jk}$  correspond with a codeword in  $D_{W_{jk\ell'}}$  with in the  $j^{\text{th}}$  position the value  $(\vec{d}_j)_{\ell'}$ , where  $(\vec{d}_j)_{\ell'} \neq (\vec{d}'_j)_{\ell'}$  since otherwise  $(\vec{d}'_j)_{W_{jk\ell'}} \in D_{W_{jk\ell'}}$ . Since  $(\vec{d}'_j)_{\ell'}$  is an entry of  $E_j$  by definition, the lemma follows.  $\square$

At first sight, it may seem that  $\mathcal{A}$  can deduce information from the minimum collision indices  $\ell_{jk}$  that are broadcast during Protocol B. However, the lemma below shows that this is not the case.

LEMMA 6. The values  $\ell_{jk}$  in Protocol A completely depend on the actions of  $\mathcal{A}$  in the first phase. In particular, these values are known to  $\mathcal{A}$  even before they are broadcast in the second phase.

*Proof.* By Lagrange's theorem, a unique linear relation  $\sum_{i=1}^{t+1} \lambda_i d_i = d_{t+2}$  necessarily holds for any codeword  $(d_1, \dots, d_{t+2}) \in D_{\mathcal{A}}$  with  $|A| = t+2$ , where the  $\lambda_i$ 's are publicly known constants that only depend on the a priori fixed evaluation points  $x_1, \dots, x_n$  for the  $(t+1)$ -dimensional Reed-Solomon code  $D$ . Lets assume that the first  $e \leq t$  coefficients are replaced by values  $d'_i$ . It is straightforward to verify that  $(d'_1, \dots, d'_{t+2}) \in D_{\mathcal{A}}$  if and only if  $\sum_{i=1}^e \lambda_i d'_i + \sum_{i=e+1}^{t+1} \lambda_i d_i = d_{t+2}$ . This is the case if and only if  $\sum_{i=1}^e \lambda_i (d_i - d'_i) = 0$ . However, the values  $d_i - d'_i$  are selected by and known to  $\mathcal{A}$ . Therefore,  $\mathcal{A}$  already knows beforehand whether any particular received partial codeword of length  $t+2$  is in the corresponding restriction of  $D$  and can in particular predict all the minimum indices  $\ell_{jk}$ .  $\square$

PROPOSITION 3. If  $(\ell, j, (\vec{a}_j)_\ell, (\vec{b}_j)_\ell) \in \mathcal{C}_i$ , then  $i \in I_{\mathcal{A}}$  and/or  $j \in I_{\mathcal{A}}$ . Furthermore, both  $(\vec{a}_j)_\ell$  and  $(\vec{b}_j)_\ell$  were already known to  $\mathcal{A}$  at the end of the first phase.

Assume that  $i \notin \mathcal{A}$ . Since the first phase of  $\hat{\Pi}_i$  is a parallel version of the first phase of  $\Pi_i$ , it is clear that  $\mathcal{A}$  obtains no information about the values  $(\vec{d}_\ell)_i$  in the first phase.

The following lemma states that the adversary does not learn anything new about the values of the entries of  $\mathcal{Z}_i$  (provided that  $\mathcal{Z}_i$  has any entries at all) in the second phase of  $\hat{\Pi}_i$ . This shows that the proposed protocol is perfectly private.

LEMMA 7. *If  $i \notin I_{\mathcal{A}}$  and  $\mathcal{Z}_i$  contains a nonzero number of entries, then  $\mathcal{A}$  obtains no new information about the values of the entries of  $\mathcal{Z}_i$  in the second phase.*

*Proof.* According to Lemma 6, the indices that are transmitted during Protocol A are selected by (and therefore known to)  $\mathcal{A}$  before the execution of the second phase. Furthermore, the values that are transmitted during Protocol A are completely uncorrelated with the vector  $\mathcal{Z}_i$  since the corresponding vectors are discarded before the vector  $\mathcal{Z}_i$  is constructed. Finally, Proposition 3 shows that only information that is already known to  $\mathcal{A}$  is transmitted during Protocol B. Therefore,  $\mathcal{A}$  does not learn anything new about the values of the entries of  $\mathcal{Z}_i$  in the second phase.  $\square$

### 5.4.5 Complexity Analysis

Choose a field  $\mathbb{F}_q$  such that its elements can be represented using bit strings of length  $\Omega(\log(n))$  and assume that  $m$  is such that  $m > nw \log_n(m)$ . The length  $\max\{0, m - nw\}$  of the vectors  $\mathcal{Z}_i$  can be chosen to be of size  $\geq cm$  for any constant  $c$  in the interval  $(0, 1)$ , by enlarging  $m$  as necessary. As Section 5.1.3 shows, this implies that we can obtain a secret key of size  $(t + 1)cm = \Omega(nm)$ , i.e., of  $\Omega(mn \log(n))$  bits. In order to have a PSMT-protocol with linear communication complexity, the total number of shares transmitted in each round should be  $O(mn^2)$  or, stated equivalently, the total number of bits transmitted in each round should be  $O(mn^2 \log(n))$ . Let us now analyze the communication complexity of the parts of the new protocol.

FIRST ROUND. For every  $i$ ,  $\mathcal{R}$  sends  $mn$  elements over channel  $i$  and  $m$  elements over every other channel  $j$ . This sums up to a total of  $O(mn)$  shares that are sent over each channel and therefore to  $O(mn^2)$  shares in total that are transmitted in the first round.

PROTOCOL: CLASSIFY CHANNELS. For every value  $i \in \{1, 2, \dots, n\}$ , at most  $O(w)$  indices and field elements are broadcast at the end of Protocol A for every  $j$ , so that in total  $O(n^2 w (\log(m) + \log(n)))$  bits have to be broadcast. This gives  $O(n^3 w (\log(m) + \log(n)))$  bits that are transmitted during Protocol A. Our assumption implies that  $m > n$ , so that this can be rewritten to  $O(n^3 w \log(m))$ . Furthermore, by assumption  $m > nw \log_n(m)$ , so that  $m \log(n) > nw \log(m)$ , i.e.,  $n^3 w \log(m) = O(mn^2 \log(n))$ .

PROTOCOL: GATHER RECONCILIATION INFORMATION. Assume that we regroup the sets  $\mathcal{C}_i$  during Protocol B as described in 5.3 and obtained the sets  $X_\ell$  with  $\ell = 1, 2, \dots, m$ . Then, using some appropriate padding between the vectors encoding sets  $X_i$ , all information can be transmitted by communicating only  $O(mn^2)$  shares.

This completes the analysis.

## 5.5 Computationally Efficient PSMT with Optimal Overhead

In 2008, Kurosawa and Suzuki [51] have published a protocol that matches the optimal communication overhead achieved by Agarwal, Cramer and de Haan [1], but that is additionally computationally efficient. After studying their techniques, it turns out that this is due to one key idea that allows to detect all modifications in the first phase while keeping the total communication low. In fact, one can use this idea to construct a protocol  $\Pi'_i$  that fits in with the general framework introduced in [1]. When one uses this new protocol to replace the protocol  $\hat{\Pi}_i$  from Section 5.4.2, one indeed obtains an efficient perfectly secure message protocol with optimal communication overhead. The idea behind it is as follows.

Let  $\mathbb{F}_q$  be a finite field and  $C \subset \mathbb{F}_q^n$  be a code of length  $n$ . When  $m$  codewords  $\vec{c}_1, \vec{c}_2, \dots, \vec{c}_m \in C$  are transmitted by  $\mathcal{R}$ ,  $\mathcal{S}$  receives vectors  $\vec{c}'_1, \vec{c}'_2, \dots, \vec{c}'_m \in \mathbb{F}_q^n$  where every  $\vec{c}'_j$  can uniquely be written as  $\vec{c}'_j = \vec{c}_j + \vec{e}_j$  with  $\vec{e}_j \in \mathbb{F}_q^n$  and  $d(\vec{e}_j) \leq t$ . Since  $\mathcal{A}$  can at most select  $t$  indices on which to alter data, the vectors  $\vec{e}_j$  span a subspace  $E \subset \mathbb{F}_q^n$  of at most dimension  $t$ .

In particular, if the minimum distance of  $C$  is larger than  $t$ ,  $E \cap C = \emptyset$  and the dimension of  $E/C$  is also  $t$ . Since  $\vec{c}'_j \equiv \vec{e}_j \pmod{C}$ , any subset of the vectors  $\vec{c}'_j$  that is maximally independent modulo  $C$  forms a base of  $E/C$ . Lemma 8 demonstrates that any such subset of vectors  $\vec{c}'_j$  then corresponds with a subset of vectors  $\vec{e}_j$  that forms a basis for  $E$ , which can be determined by  $\mathcal{R}$  upon receiving these vectors  $\vec{c}'_j$ . This suffices to ensure that  $\mathcal{R}$  can detect all channels with modifications.

The details of the protocol  $\Pi'_i$  are as follows. Let  $\mathbb{F}_q$  be such that  $|\mathbb{F}_q| > n + t$  and let  $\mathcal{N} = \{1, 2, \dots, n\}$ . The first round is just as in the protocol  $\hat{\Pi}_i$ , where we choose  $m = o(n^2)$ . Again, let  $C_i \subset \mathbb{F}_{q^{t+1}}$  be the  $i$ -concentrated  $[n, t + 1]$  Reed-Solomon code and  $D_i$  be the Reed-Solomon code obtained by replacing every  $i^{\text{th}}$  coefficient  $\alpha$  by  $f_\alpha(x_i)$ .

### 5.5.1 Round One

In the first round,  $\mathcal{R}$  first selects  $n$  codewords  $\vec{c}_1, \vec{c}_2, \dots, \vec{c}_m \in C_i$  uniformly at random.  $\mathcal{R}$  then proceeds by transmitting  $\vec{c}_1, \vec{c}_2, \dots, \vec{c}_m$  as usual; by transmitting the respective values  $\{(\vec{c}_k)_j\}_{k=1}^m$  over channel  $r_j$  for  $j = 1, 2, \dots, n$ .

### 5.5.2 Round Two

Assume that  $\mathcal{S}$  receives the vectors  $\vec{c}'_1, \vec{c}'_2, \dots, \vec{c}'_m \in \mathbb{F}_{q^{t+1}}$  and without loss of generality, we can assume that  $(\vec{c}'_k)_j \in \mathbb{F}_q$  for  $j \in \mathcal{N} \setminus \{i\}$  and any  $k \in \{1, 2, \dots, m\}$ . Let

$\vec{d}'_1, \vec{d}'_2, \dots, \vec{d}'_m$  be the corresponding vectors where the  $i^{\text{th}}$  coefficients  $\alpha'$  are replaced by  $f_{\alpha'}(x_i)$ .

$\mathcal{S}$  selects a subset  $A$  of  $\vec{d}'_1, \vec{d}'_2, \dots, \vec{d}'_n$  that is maximally independent modulo the code  $D_i$ . As described above, this subset allows  $\mathcal{R}$  to identify all channels that have had modifications and clearly  $|A| \leq t$ . The following protocol now describes the details of this procedure.

PROTOCOL A: IMPROVED CLASSIFY CHANNELS

1. For  $i = 1, 2, \dots, m$   $\mathcal{S}$  iteratively adds vectors  $\vec{d}'_i$  to (an initially empty set)  $A$  whenever adding the vector  $\vec{d}'_i$  does not make the vectors in  $A$  dependent modulo  $D_i$ . Whether a set of vectors is dependent modulo  $D_i$  can be verified at every step using the technique described in Section 5.5.3. This results in a set  $A$  that is maximally independent modulo the code  $D_i$ .
2.  $\mathcal{S}$  broadcasts  $A$ . Furthermore,  $\mathcal{S}$  defines

$$\mathcal{Z}_i := \{f_{(\vec{c}'_l)_i}(x_i) | \vec{c}'_l \notin E\}.$$

In parallel,  $\mathcal{S}$  executes Protocol B from Section 5.4.2.

Upon receiving the set  $E$ ,  $\mathcal{R}$  can determine the error vectors, and disqualify all channels in their support. Then, using Protocol C from Section 5.4.2  $\mathcal{R}$  can recover the sets  $\mathcal{Z}_i$  as in the previous protocol.

### 5.5.3 Checking for Dependence Modulo $C$

Given vectors  $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_k$ , one can efficiently check whether they are dependent modulo  $C$  as follows. Assume that they are, so that there exist values  $\alpha_1, \alpha_2, \dots, \alpha_k \in \mathbb{F}_q$  for which  $\sum_{i=1}^k \alpha_i \vec{b}_i \equiv \vec{0} \pmod{C}$ . Then this implies that  $\vec{c} = (c_1, c_2, \dots, c_n) = \sum_{i=1}^k \alpha_i \vec{b}_i \in C$ . Since  $C$  is a Reed-Solomon code of dimension  $t + 1$ , we know that  $\vec{c} \in C$  if and only if the last  $n - (t + 1)$  coefficients are such that  $c_j = \sum_{i=1}^{t+1} \lambda_i^{(j)} c_i$  for  $j = t + 2, t + 3, \dots, n$ , where the constant coefficients  $\lambda_i^{(j)}$  are derived from Lagrange's interpolation formula.

Since  $c_j = \sum_{i=1}^k \alpha_i b_{ij}$  for  $j = 1, 2, \dots, n$ , where  $\vec{b}_i = (b_{i1}, b_{i2}, \dots, b_{in})$  for  $i = 1, 2, \dots, k$ , we obtain  $n - (t + 1)$  equations in at most  $t$  variables. Now the vectors are dependent modulo  $C$  if and only if this system of equations has a solution, which can be determined in time polynomial in  $n$ .

It remains to show that any set of vectors  $\vec{d}'_1, \dots, \vec{d}'_k$  that is selected in the protocol leads to a basis  $\vec{e}_1, \dots, \vec{e}_k$  of the space  $E$ .

LEMMA 8. *Let  $k \leq t$ . Then  $k$  vectors  $\{\vec{c}'_{i_1}, \vec{c}'_{i_2}, \dots, \vec{c}'_{i_k}\}$  that are received by  $\mathcal{S}$  are independent modulo  $C$  if and only if the corresponding error vectors in the set  $\mathcal{E} = \{\vec{e}_{i_1}, \vec{e}_{i_2}, \dots, \vec{e}_{i_k}\}$  are independent.*

*Proof.* We will demonstrate that the vectors  $\{\vec{c}_{i_1}, \vec{c}_{i_2}, \dots, \vec{c}_{i_k}\}$  are dependent modulo  $C$  if and only if the corresponding error vectors in  $\mathcal{E}$  are dependent. Note that one direction here is trivial, i.e., if the error vectors in  $\mathcal{E}$  are dependent then the vectors  $\{\vec{c}_{i_1}, \vec{c}_{i_2}, \dots, \vec{c}_{i_k}\}$  are dependent modulo  $C$ .

Assume that  $\vec{c}_{i_1} \equiv \sum_{v=2}^k \lambda_v \vec{c}_{i_v} \pmod{C}$  for some  $\lambda_2, \dots, \lambda_k \in \mathbb{F}_q$ . Then  $\vec{e}_{i_1} \equiv \sum_{v=2}^k \lambda_v \vec{e}_{i_v} \pmod{C}$  and we show that this implies that  $\vec{e}_{i_1} = \sum_{v=2}^k \lambda_v \vec{e}_{i_v}$ . To simplify notation, denote  $\vec{a}_1 = \vec{e}_{i_1}$  and  $\vec{a}_2 = \sum_{v=2}^k \lambda_v \vec{e}_{i_v}$  and note that  $d(\vec{a}_1, \vec{a}_2) \leq t$ . Then since  $\vec{a}_1 \equiv \vec{a}_2 \pmod{C}$ , there exist two codewords  $\vec{b}_1, \vec{b}_2 \in C$  such that  $\vec{b}_1 + \vec{a}_1 = \vec{b}_2 + \vec{a}_2$ . This implies that  $\vec{a}_2 - \vec{a}_1 = \vec{b}_1 - \vec{b}_2$ . However, since  $\vec{b}_1 - \vec{b}_2 \in C$ ,  $d(C) > t$  and  $d(\vec{a}_2 - \vec{a}_1) \leq t$ , it follows that  $\vec{a}_1 = \vec{a}_2$ .  $\square$

## 5.6 Extra Phases Do Not Improve Efficiency

Since communication-optimal protocols exist when one allows exactly two communication phases, the next natural question to ask is whether one can get an even lower communication overhead by allowing additional phases. It turns out that this is not the case, as proven in 2007 by Srinathan, Prasad and Pandu Rangan [71]. In their paper they achieve a general lower bound via a reduction of perfectly secure message transmission protocols to secret sharing schemes with error detection. We give a simplified overview of this result that is sufficient for our communication model.

**THEOREM 21.** ([71]) *Any perfectly secure message transmission protocol for  $n > 2t$  channels requires communication overhead  $\geq n/(n - 2t)$ .*

*Proof.* Consider the data transmitted during an execution of some perfectly secure message transmission for some message  $M \in \mathcal{M}$ , where the adversary reads the data transmitted over up to  $t$  channels, but forwards data unmodified. We claim that this data has the following two properties:

1. The data transmitted on any  $t$  channels is independent of the message  $M$ .
2. The data transmitted on any  $n - t$  channels determines the message  $M$ .

The first property is an obvious consequence of the privacy property of the protocol. Assume now that the second property does not hold. Then there exists some message  $M' \in \mathcal{M}$  and some collection of data for the  $t$  channels that adversary controls such that the data transmitted on all  $n$  channels is both consistent with the message  $M$  and the message  $M'$ . This implies that there exists a protocol execution after which  $\mathcal{R}$  cannot distinguish between these two cases at the end of the protocol, which contradicts the correctness property of the protocol. Therefore, the second property must also hold.

The bound now follows from a similar argument to that made in Theorem 19.  $\square$

## 5.7 Three-Phase PSMT

Surprisingly, despite the fact that using more than two communication phases does not allow to improve the communication overhead (or computational efficiency), there is indeed motivation to consider three-phase perfectly secure message transmission protocols in the setting where  $n \geq 2t + 1$ . This is due to the fact that no perfectly secure one-phase protocols exist for this setting, while it could be preferable to use one communication phase instead of two communication phases in certain situations.

In this section we demonstrate that some three-phase protocols essentially become one-phase protocols when  $\mathcal{A}$  does not modify any data at all in the first phase, which gives these types of protocols an advantage when this is likely to occur often. Motivation for such adversarial behavior could be for instance the situation where gaining control of a sequence of channels  $r_j, s_j$  is very costly, making it important that the intrusion is not detected until it is needed to disturb an important communication. Note that the best known protocols can in fact guarantee detection of all adversarial behavior, so that this is likely to present a very practical problem. As an added bonus, these three-phase protocols still guarantee perfect correctness, perfect privacy and detection of all modifications when the adversary does modify data, at the cost of two extra communication phases.

Concretely, we demonstrate that three-phase protocols can be organized in such a way that  $\mathcal{R}$  can immediately determine the message  $M$  after the first phase when no data is modified.<sup>3</sup> Additionally, we propose a modification with respect to the protocol  $\Pi_i$ , which is used here, that lowers the amount of communication required when the remainder of the three-phase protocol is continued after the first phase.

### 5.7.1 Basic Protocol

The first efficient three-phase protocol, which has been described in [64, 19], is in essence a simple extension of the two-phase protocol due to Sayeed and Abu-Amara [64] from Section 5.1. Conceptually, the third phase is used to “repair” all the values that were incorrectly received at the end of the first phase. The most basic version of the protocol works as follows.

Initially, protocol  $\Pi_i$  is executed in parallel for every index  $i \in \{1, 2, \dots, n\}$  with the modification that  $\mathcal{S}$  and  $\mathcal{R}$  swap roles. This results in  $n$  random values  $\{v_1, v_2, \dots, v_n\}$  that are transmitted by  $\mathcal{S}$ , of which at least  $t + 1$  are correctly received by  $\mathcal{R}$ . Furthermore,  $\mathcal{S}$  finds out in the second phase which values were correctly received and broadcasts the values that were not correctly received to  $\mathcal{R}$  in the third round.

<sup>3</sup>Alternatively, the first phase of such a three-phase protocol can be executed in parallel with the first phase of a two-phase protocol to achieve the same effect.

$\mathcal{R}$  now knows all the correct initially transmitted values. Also,  $\mathcal{A}$  knows at most  $t$  of the values received by  $\mathcal{R}$ , which correspond to the channels that are under his control.  $\mathcal{S}$  and  $\mathcal{R}$  can now apply the privacy amplification technique from Section 5.1.3 to obtain  $t + 1$  completely secret elements  $v_{n+1}, v_{n+2}, \dots, v_{n+t+1}$ , which can then be used as a one-time pad in the third phase to mask and transmit a message of size  $t + 1$  elements.

As noted earlier, the parallel application of the protocol  $\Pi_i$  above can be performed efficiently when combined with the low communication reliable transmission technique for the collisions that are transmitted in the second round. This gives the protocol described above linear communication overhead.

### 5.7.2 Expected One-Phase PSMT for $n \geq 2t + 1$

The idea is to encode the message within the  $n$  values  $v_i$  that were previously used to generate a random shared secret key. Before, since at most  $t$  of these values were leaked during the protocol, privacy amplification allowed for the extraction of  $t + 1$  secret random values. Now we specify how to use this “ $t + 1$  secrecy” to encode a message consisting of  $t + 1$  elements.

It should be the case that  $n$  values suffice to reconstruct the message, while  $t$  values should give no information about the  $(t + 1)$ -element message. We use an  $[n + t + 1, n - 1]$  Reed-Solomon code for this, where we select a codeword at random under the restriction that the last  $t + 1$  coefficients match the respective parts of the message. The rest of the protocol works the same as before using the first  $n$  coefficients for transmission. It is straightforward to verify that the requirements for PSMT are satisfied.

If there are now no modifications in the first phase, all coefficients are received correctly by  $\mathcal{R}$  at the end of the first phase and this is visible to  $\mathcal{R}$  due to the lack of collisions to transmit in the the second phase.  $\mathcal{R}$  can then immediately decode the message and the transmission is basically done. In certain settings,  $\mathcal{R}$  might have to notify  $\mathcal{S}$  that this is the case, which would add another (low-communication) phase to the protocol.

### 5.7.3 Protocol $\Pi'_i$

We can slightly improve the total amount of communication of the three-round protocol in the previous section. The improvement technique is generic and can also be applied to most of the other interactive PSMT protocols that are described in this paper, so we list it for completeness. The idea is to replace some of the shares that are transmitted in the second round, i.e., the ones that are used only for the detection of corruptions, by a relevant bit in their representation.

The easiest way to demonstrate the idea is by replacing the protocol  $\Pi_i$  from Section 5.1.1 by the following protocol  $\Pi'_i$ , where we again swap the roles of  $\mathcal{S}$  of  $\mathcal{R}$  so it can be used for three-phase protocols.

PROTOCOL  $\Pi'_i$

1. The first round is identical to that of  $\Pi_i$ . We denote the value that  $\mathcal{R}$  receives on channel  $r_i$  by  $\alpha'$  and the values received on the other channels  $r_j$  by  $c'_j$ . This completes the first round.
2. The changes we make are in the second round. If  $\mathcal{R}$  receives any incorrectly formed data on any of the channels,  $\mathcal{R}$  broadcasts a notification for these channels. Otherwise, for every pair of values such that  $f_{\alpha'}(x_j) \neq c'_j$ ,  $\mathcal{R}$  now broadcasts  $j$ , the index  $k$  of a bit where  $f_{\alpha'}(x_j)$  and  $c'_j$  differ together with the value of the  $k^{\text{th}}$  bit in  $f_{\alpha'}(x_j)$  (whereas in protocol  $\Pi_i$ ,  $\mathcal{R}$  used to broadcast  $j$  and the entire value  $f_{\alpha'}(x_j)$ ). Finally,  $\mathcal{S}$  verifies for all received values whether the  $k^{\text{th}}$  bit of  $f_{\alpha'}(x_j)$  equals the  $k^{\text{th}}$  bit of  $c_j$  and identifies that  $i \in I_{\mathcal{A}}$  if this is not the case or if  $\mathcal{R}$  broadcast a notification for channel  $r_i$ .

The privacy and correctness properties follow from an argument similar to that made in Section 5.1.1. So, basically, the verification of the received values can already be performed using a single bit that differs between conflicting values, which saves communication during the second round. Similar adjustments are possible with most of the other interactive protocols to improve the total amount of required communication.

## 5.8 Two-Phase PSMT for Non-Tight Parameters

We now discuss what is possible for two-phase protocols when we move slightly away from the worst-case parameters  $n = 2t + 1$  and consider  $n \geq (2 + \epsilon)t$  with  $\epsilon > 0$ . We first mention in our work [1] that one can slightly modify our communication-optimal protocol (as described in Section 5.4) for these new parameters so that it turns into a protocol with constant communication overhead, which is trivially seen to be optimal. This is due to the fact that the modified protocol allows to extract a secret key that is a linear factor (in  $n$ ) longer than in the original protocol. When one applies a similar transformation to the communication-efficient protocol of Kurosawa and Suzuki [51], the resulting protocol also introduces a constant overhead and is additionally communication-efficient.

We note that the first known communication-efficient two-phase protocol with constant communication overhead for  $n \geq (2 + \epsilon)t$  is due to Fitzi, Franklin, Garay and Harsha Vardhan [33]. This result was published in 2007, in between our 2006

publication and the publication of the paper by Kurosawa and Suzuki in 2008. Since the protocol of Fitzi et al. uses techniques that are fundamentally different from those described earlier, we consider it worthwhile to present them here for completeness.

### 5.8.1 Sketch of the Protocol

Assume that  $n \geq (2 + \epsilon)t$  and let  $N$  and  $T$  be such that  $N \geq (3 + \delta)T$ . We describe the relation of  $N$  and  $T$  to  $n$  and  $t$  later on. Similar to before, let  $\mathcal{C}$  be a  $[N, (1 + \delta)T]$  Reed-Solomon code over a sufficiently large finite field  $\mathbb{F}$ . We assume again that every codeword  $\vec{c} \in \mathcal{C}$  corresponds with a message  $M \in \mathcal{M}$ .

The idea is now to execute  $N$  (not necessarily perfectly secure) two-phase message transmission protocols in parallel, where during every execution  $\mathcal{S}$  attempts to transmit one coefficient of a codeword  $\vec{c}$  corresponding to the message  $M$ . If we can guarantee perfect security for all but at most  $T$  executions,  $\mathcal{R}$  ends up with a vector  $\vec{c} \in \mathbb{F}_q^N$  that introduces at most  $T$  errors, where at least  $N - T$  correctly received coefficients are jointly statistically independent from any data observed by  $\mathcal{A}$ . This basically shows that the resulting parallel protocol  $\Gamma$  behaves like a one-round perfectly secure message transmission protocol for parameters  $N \geq (3 + \delta)T$  where we can allow a message of size  $\delta T$ .

We will demonstrate that the message transmission protocols all introduce constant overhead, which shows that the entire protocol introduces constant overhead.

### 5.8.2 The Two-Phase Message Transmission Setup

The  $N$  message transmission protocols are set up as follows. Let  $\Pi$  be any known two-phase perfectly secure message transmission protocol for  $\nu \geq (2 + \epsilon)t$ . The idea is to execute this protocol for  $N$  distinct  $\nu$ -sized subsets of the channels, where it is allowed to select channels multiple times in a subset. The selection of subsets should be such that at most  $T$  executions can fail. Note that the execution of  $\Pi$  can only fail for subsets involving a majority of 'corrupted' channels, where we count with multiplicities when channels occur more than once in the subset.

The selection of subsets that we apply is one that ranges over all elements of  $\mathcal{N}^\nu$ . The main issue that needs to be worked out is which value of  $\nu$  should be used in order to ensure that there are at most  $T$  'failing' subsets. Since every element of  $\mathcal{N}^\nu$  is used exactly once, we can compute the fraction of failing subsets using probability theory. The idea is to compute the probability of selecting a bad subset when picking one of the subsets uniformly at random from  $\mathcal{N}^\nu$ . From this it follows which value of  $\nu$  makes this probability sufficiently low that the fraction of bad subsets is less than or equal to  $T$ .

### 5.8.3 Parameter Estimation for $\nu$

We now give a (deterministic) estimation on the ratio of corrupted virtual channels, based on the choice of  $\nu$ . This determines for which value of  $\nu$  the protocol  $\Gamma$  succeeds.

Let  $\nu$  be fixed and  $p$  be the probability that when we select an  $\nu$ -tuple of indices uniformly at random from  $\mathcal{N}^\nu$ , more than half of the indices correspond to a corrupted channel. In other words,  $p$  is the probability that the two-phase secure message transmission protocol  $\Pi$  fails when executed using the channels corresponding to a randomly selected  $\nu$ -tuple. If this probability  $p$  is at most  $\frac{T}{N} = \frac{1}{3+\delta}$  then the conceptual one-phase protocol  $\Gamma$  succeeds.

Consider the random variable  $X \in \{0, \dots, \nu\}$  that denotes the number of indices in the selection that correspond to corrupted channels. The goal is to show that there is a constant  $\nu$  such that  $p = P(X \geq \nu/2) \leq \frac{1}{3+\delta}$ , so that the number of failed executions of the protocol  $\Pi$  is at most  $T = \frac{N}{3+\delta}$ .

We can consider the variable  $X$  to be the sum of  $\nu$  independent variables

$$X_1, X_2, \dots, X_\nu,$$

where the variable  $X_i \in \{0, 1\}$  denotes whether the  $i^{\text{th}}$  uniformly randomly selected channel from the set of channels  $\mathcal{N}$  can be read and modified by  $\mathcal{A}$ . This allows to bound the probability  $p$  using the Chernoff bound.

**THEOREM 22.** (Chernoff bound [44], taken from [33]) *Let  $X_i$  ( $1 \leq i \leq n$ ) be a sequence of independent 0-1 distributed random variables with expected value  $\mu$ . By  $\mathcal{C}(\mu, n, \lambda)$  ( $\lambda > 1$ ) we denote the probability that, out of  $n$  trials, the outcome exceeds the expected value  $n\mu$  by a given factor depending on  $\lambda$ . The following inequality, which holds for  $1 < \lambda < 2e$ , bounds this probability.*

$$\mathcal{C}(\mu, n, \lambda) = P\left(\sum_{i=1}^n X_i \geq \lambda \mu n\right) \leq e^{-\frac{\mu n (\lambda - 1)^2}{2}}. \quad (5.1)$$

Using the Chernoff bound, it follows that

$$P\left(X \geq \frac{\nu}{2} = \lambda \mu \nu = \lambda \frac{\nu}{2 + \epsilon}\right) \leq e^{-\frac{\nu}{2(2+\epsilon)}(\lambda-1)^2},$$

where  $\lambda = \frac{2+\epsilon}{2}$ . Note that for the claim to hold it is required that

$$e^{-\frac{\nu}{2(2+\epsilon)}(\lambda-1)^2} \stackrel{!}{\leq} \frac{1}{3+\delta}.$$

It follows that it suffices that

$$\nu \geq \left\lceil \frac{8 \ln(3+\delta)(2+\epsilon)}{\epsilon^2} \right\rceil,$$

## 5.8. Two-Phase PSMT for Non-Tight Parameters

---

obtaining a lower-bound estimation on  $\nu$  depending on constants  $\epsilon$  and  $\delta$ , where  $\epsilon$  is an input parameter and  $\delta$  is any positive constant of free choice.

Since  $\nu$  is a constant here, the protocol  $\Pi$  executed on any subset of  $\nu$  channels is easily seen to involve constant communication and therefore introduces at most constant communication overhead. From this it follows that the protocol  $\Gamma$  gives constant overhead as well.



## Part III

# Ramp Sharing



## Chapter 6

# Ramp Schemes

Linear secret sharing is considered a fundamental primitive of any unconditionally secure multi-party computation protocol. In particular, there exist efficient general techniques to construct multi-party computation protocols secure against a passive adversary from multiplicative secret sharing schemes. Against an active adversary, one can efficiently obtain perfectly secure protocols from strongly multiplicative secret sharing schemes or, when one allows for an error probability, statistically secure protocols from multiplicative secret sharing schemes.

As a consequence of these results, the design of secure multi-party computation protocols comes down to the design of (strongly) multiplicative secret sharing schemes. Since the resulting multi-party computation protocols make heavy use of the secret sharing functionality of the underlying scheme, the amount of communication involved in the protocol is strongly linked to the cost of secret sharing, which in terms of communication consists of the transmission of the shares in the scheme to the participants.

However, there exist important limits on the sizes of these shares. Most importantly, since the secret corresponds with a single row in the secret sharing matrix and the shares correspond with one or multiple rows, every share is at least as large as the secret. This implies an immediate lower bound on the amount of required communication in terms of the size of the secret. Furthermore, this restriction is independent of the adversary structure of the underlying linear secret sharing scheme.

Although linear secret sharing schemes exist for any monotonous adversary structure, perhaps the most natural and most commonly used adversary structure is the  $t$ -threshold adversary structure that rejects all subsets of size up to a given value  $t \in \mathbb{Z}_{\geq 0}$ . For linear secret sharing schemes with these adversary structures, the  $t$ -threshold secret sharing schemes, strong limitations on the share sizes are known.

We recall the following two important limitations on  $t$ -threshold schemes from

## Section 2.5.5.

1. Each share is at least as large as the secret.
2. For any ideal  $t$ -threshold scheme with  $n$  participants, the finite field  $\mathbb{F}_q$  over which it is defined needs to be of size at least  $(n-2)/2$  if  $1 \leq t \leq n-3$ , while it is conjectured that this lower bound is in fact  $n-1$ .

Note that the first limitation mentioned here also holds more generally for perfect secret sharing schemes without dummy indices. In particular, ideal threshold secret sharing schemes have the additional restriction that the size of the secret, and therefore also the size of any share, is at least linear in the number of participants. While it is possible for non-ideal schemes to use fields of size much smaller than  $n$ , the minimum average share size for these schemes is larger than  $\log_q(n/2)$ , which is even beyond that required for ideal threshold schemes.

In the sequel, we construct more efficient (strongly) multiplicative secret sharing schemes that circumvent these restrictions. To be more precise, we design secret sharing schemes that are “almost” threshold, in the sense that for two values  $t$  and  $r$  all sets of cardinality at most  $t$  are rejected and all sets of cardinality at least  $r$  are accepted, while we leave open what happens when one considers sets with cardinality between these two thresholds. It is precisely the fact that these schemes are non-perfect that allows us to obtain better results, although we manage to keep the “gap” between  $t$  and  $r$  restricted to at most a (small) constant fraction of the number of participants.

For both of the limitations listed we present appropriate solutions. We design high information rate schemes, which can have shares that are (much) smaller than the secret, and schemes that work over fields that are much smaller than the number of participants or even constant-sized. High information rate schemes have been known in the literature for some time, in particular based on packing larger secrets in the secret sharing polynomial used for Shamir’s secret sharing scheme, although multiplicativity for such schemes has previously only been considered in work of Franklin and Yung [34]. Schemes over small fields were on the other hand only recently introduced in a 2006 paper by Chen and Cramer [15].

In this chapter we define a general framework for such schemes, which we in their most general form call *ramp schemes*. We then proceed by defining the notions of multiplication and strong multiplication for ramp schemes, which requires some special care since multiplications for these schemes involve vectors instead of elements. Furthermore, we list some conditions on the rejected sets that need to hold in order for multiplication or strong multiplication to be possible at all.

In the next chapters we then design ramp schemes that tackle the two limitations using two distinct technical approaches. In Chapter 7 we introduce a new connection

between ramp schemes and codes with their dual code. This leads to a method for constructing almost-threshold ramp schemes that is fundamentally different from the only previously known methods based on polynomial evaluation. In particular, we show that we can use this method to construct families of multiplicative ramp schemes over constant-sized fields. We note that the latter result is similar to that of Chen and Cramer. However, unlike their schemes, which can be hard to construct due to their heavy use of algebraic-geometric building blocks, our schemes are easy to construct and can be based on arbitrary or even randomly selected codes. However, it is worth noting that one can only obtain multiplicative ramp schemes in this manner.

On the other hand, the theoretical significance of the schemes of Chen and Cramer lies in the fact that they generalize the construction of all previously known almost-threshold ramp schemes with strong multiplication. In Chapter 8 we describe their algebraic-geometric schemes and then use the techniques involved to introduce a new general class of strongly multiplicative almost-threshold ramp schemes.

The material in this chapter and the following chapters is based on [17], [21] and [16].

## 6.1 Definition

In a nutshell, ramp schemes generalize the notion of linear secret sharing scheme, as introduced in Section 2.5.2, by allowing the secret to be a vector over  $\mathbb{F}_q$  rather than just a single element from the finite field  $\mathbb{F}_q$ . We define a ramp scheme over the field  $\mathbb{F}_q$  as follows.

**DEFINITION 32.** *A ramp scheme is a triple  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$ , where  $\mathbb{F}_q$  is a finite field,  $M \in \mathbb{F}_q^{(d+k) \times e}$  is a matrix with as its first  $k$  rows the unit vectors  $\vec{e}_1, \vec{e}_2, \dots, \vec{e}_k \in \mathbb{F}_q^e$  and  $\phi : \{1, 2, \dots, d\} \rightarrow \mathcal{N}$  is a surjective function. The size of  $\mathcal{M}$  is  $d$  and the information rate of  $\mathcal{M}$  is  $k/d$ .*

Label the last  $d$  rows of the matrix by  $1, 2, \dots, d$  and let  $M_i$  denote the row of  $M$  labeled by  $i$ . For any subset  $B \subset \mathcal{N}$ , we let  $M_B$  denote the submatrix consisting of the rows  $\{M_i\}_{i \in \psi^{-1}(B)}$ . Furthermore, let  $\text{Im}(M_B^T)$  denote the  $\mathbb{F}_q$ -linear span of the rows of  $M_B$  and  $\text{Ker}(M_B)$  consist of the vectors  $\vec{\kappa} \in \mathbb{F}_q^e$  such that  $M_B \cdot \vec{\kappa} = \vec{0}$ .

**DEFINITION 33.** *For a ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$*

- *a set  $B \subset \mathcal{N}$  is accepted if the  $i^{\text{th}}$  unit vector  $\vec{e}_i \in \mathbb{F}_q^e$  is in the image of  $M_B^T$  for all  $i \in \{1, \dots, k\}$ .*
- *a set  $B \subset \mathcal{N}$  is rejected if for any  $k$  elements  $w_1, \dots, w_k \in \mathbb{F}_q$  there exists a vector  $\vec{v} \in \text{Ker} M_B$  such that its first  $k$  coordinates are  $w_1, \dots, w_k$ .*

It can be shown that a ramp scheme is indeed a secret sharing scheme using techniques very similar to those in Theorem 13. The proof of the statement is rather straightforward and omitted here. Additionally note that, unlike linear secret sharing schemes, ramp schemes can be (and in fact usually are) non-perfect.

Note that the definition of a ramp scheme with  $k = 1$  is equivalent to the standard definition of a linear secret sharing scheme. For linear secret sharing schemes, the information rate is maximal when the scheme is ideal, in which case the information rate is  $1/n$ . We say that ramp schemes that can achieve an information rate larger than  $1/n$  have *high information rate*.

## 6.2 (Strongly) Multiplicative Ramp Schemes

We now generalize the multiplication and strong multiplication properties for linear secret sharing schemes. Since the secrets in ramp schemes are vectors rather than single values, and we would like to keep the definition as general as possible, we do not assume a standard definition for the multiplication of two vectors. Therefore, multiplication properties for ramp schemes also necessarily include a definition for the relevant multiplication map.

In the following, let  $\odot : \mathbb{F}_q^k \times \mathbb{F}_q^k \rightarrow \mathbb{F}_q^k$  be a symmetric non-degenerate bilinear map. We define multiplication of secret vectors  $\vec{s}, \vec{t} \in \mathbb{F}_q^k$  to be via this map, which we denote by  $\vec{s} \odot \vec{t}$ . Furthermore, for  $i \in \{1, 2, \dots, k\}$  let  $\nu_i$  denote the  $i^{\text{th}}$  unit vector in  $\mathbb{F}_q^k$ .

We first repeat some notation from Section 2.5.3. For any two vectors  $\vec{x} = (x_1, x_2, \dots, x_e), \vec{y} = (y_1, y_2, \dots, y_e) \in \mathbb{F}_q^e$ , let  $\vec{x} \otimes \vec{y}$  denote the vector

$$(x_1 \cdot y_1, x_2 \cdot y_2, \dots, x_n \cdot y_n) = (x_1 y_1, x_1 y_2, \dots, x_e y_{e-1}, x_e y_e) \in \mathbb{F}_q^{e^2}.$$

Let  $V_i \subset \mathbb{F}_q^e$  denote the subspace spanned by the row vectors of the matrix  $M_{\{i\}}$  and  $\hat{V}_i$  denote the subspace  $V_i \otimes V_i \subset \mathbb{F}_q^{e^2}$  spanned by all vectors  $\vec{x} \otimes \vec{y}$  with  $\vec{x}, \vec{y} \in V_i$ . Furthermore, let  $\hat{V}_B$  denote the subspace of  $\mathbb{F}_q^{e^2}$  spanned by all vectors in the spaces  $\{\hat{V}_i\}_{i \in B}$ .

**DEFINITION 34.** A ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$  is multiplicative under the multiplication map  $\odot$  if for any  $i, j \in \{1, 2, \dots, k\}$  such that  $\vec{v}_i \odot \vec{v}_j \neq \vec{0}$  it holds that

$$\vec{\epsilon}_i \otimes \vec{\epsilon}_j \in \hat{V}_N.$$

**DEFINITION 35.** A ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$  is  $\mathcal{A}$ -strongly multiplicative under the multiplication map  $\odot$  if the following two conditions hold.

1.  $\mathcal{M}$  rejects all sets in  $\mathcal{A}$ .

## 6.2. (Strongly) Multiplicative Ramp Schemes

---

2. For any set  $B \in \mathcal{A}$ ,  $\mathcal{M}$  is multiplicative under the multiplication map  $\odot$  with respect to the set  $C = \mathcal{N} \setminus B$ , i.e., for any  $i, j \in \{1, 2, \dots, k\}$  such that  $\vec{v}_i \odot \vec{v}_j \neq \vec{0}$  it holds that

$$\vec{e}_i \otimes \vec{e}_j \in \hat{V}_C.$$

If  $d = n$  and the map  $\psi$  is a bijection we can, similar to what we did for linear secret sharing schemes, replace Definitions 34 and 35 with convenient equivalent definitions.

Let  $\mathcal{M} = (\mathbb{F}_q, M, Id_{\mathcal{N}})$  be a ramp scheme and consider the experiment where a vector  $\vec{b} = (b_1, b_2, \dots, b_e) \in \mathbb{F}_q^e$  is selected uniformly at random. The vector  $(b_1, b_2, \dots, b_k)$  is called the *secret* and the elements  $M_{\{i\}} \cdot \vec{b}$  for  $i \in \mathcal{N}$  are called the *shares*.

**DEFINITION 36.** Let  $d = n$ . A ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, Id_{\mathcal{N}})$  is multiplicative under the multiplication map  $\odot$  if there exist vectors  $\vec{\lambda}_j = (\lambda_j^{(1)}, \dots, \lambda_j^{(k)}) \in \mathbb{F}_q^k$  for  $j = 1, \dots, n$  such that for any two secrets  $\vec{s}$  and  $\vec{s}'$  with respective shares  $s_1, s_2, \dots, s_n$  and  $s'_1, s'_2, \dots, s'_n$  we have that

$$\vec{s} \odot \vec{s}' = \sum_{j=1}^n \vec{\lambda}_j s_j s'_j.$$

**DEFINITION 37.** Let  $d = n$ . A ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, Id_{\mathcal{N}})$  is  $\mathcal{A}$ -strongly multiplicative under the multiplication map  $\odot$  if

1.  $\mathcal{M}$  rejects all sets in  $\mathcal{A}$ .
2. For any set  $B \in \mathcal{A}$ ,  $\mathcal{M}$  is multiplicative with respect to the set  $C = \mathcal{N} \setminus B$ , i.e., given any set  $B \in \mathcal{A}$  there exist  $\{\vec{\lambda}_j\}_{j \in C}$  in  $\mathbb{F}_q^k$  such that for any two secrets  $\vec{s}$  and  $\vec{s}'$  with respective shares  $s_1, s_2, \dots, s_n$  and  $s'_1, s'_2, \dots, s'_n$  we have that

$$\vec{s} \odot \vec{s}' = \sum_{j \in C} \vec{\lambda}_j s_j s'_j.$$

Since these definitions only depend on the secrets and the shares originating from the ramp scheme, we in the sequel restrict to describing how the secrets and shares are selected when defining our ramp schemes and will often omit the explicit description of the matrix  $M$ .

Furthermore we remark that, although whether a set is accepted or rejected is formally defined in terms of the matrix  $M$ , ramp sharing based on  $M$  is consistent with the general definition of secret sharing found in Definition 19. Therefore, it suffices to argue about the correlation between the secret and the shares corresponding to a set in order to determine whether a set is accepted or rejected.

### 6.3 Conditions for (Strong) Multiplicativity

For linear secret sharing schemes it is well-known that if the scheme is multiplicative (strongly multiplicative) then the set of all rejected sets is  $Q^{(2)}$  ( $Q^{(3)}$ ). We now demonstrate the equivalent conditions for ramp schemes.

Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be a monotone Boolean function and  $\mathcal{A}$  and  $\Gamma$  be such that  $\mathcal{A} \subset \mathcal{A}_f$  and  $\Gamma \subset \Gamma_f$ .

DEFINITION 38. *Let  $\mathcal{M}$  be a ramp scheme. The adversary structure  $\mathcal{A}_{\mathcal{M}}$  of  $\mathcal{M}$  is the set consisting of all sets that are rejected by  $\mathcal{M}$ . The access structure  $\Gamma_{\mathcal{M}}$  of  $\mathcal{M}$  is the set consisting of all sets that are accepted by  $\mathcal{M}$ .*

DEFINITION 39. *A tuple  $(\mathcal{A}, \Gamma)$  is  $R^{(2)}$  if there do not exist  $B_1 \in \mathcal{A}$  and  $B_2 \subset \mathcal{N}$  with  $B_2 \notin \Gamma$  such that  $B_1 \cup B_2 = \mathcal{N}$ . A tuple  $(\mathcal{A}, \Gamma)$  is  $R^{(3)}$  if there do not exist  $B_1, B_2 \in \mathcal{A}$  and  $B_3 \subset \mathcal{N}$  with  $B_3 \notin \Gamma$  such that  $B_1 \cup B_2 \cup B_3 = \mathcal{N}$ .*

PROPOSITION 4. *Suppose there exists a multiplicative ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$ , with respect to some multiplication map  $\odot$ , that has access structure  $\Gamma$  and adversary structure  $\mathcal{A}$ . Then the tuple  $(\mathcal{A}, \Gamma)$  is  $R^{(2)}$ .*

PROOF. Suppose that the tuple  $(\mathcal{A}, \Gamma)$  is not  $R^{(2)}$ , i.e., there exists a set  $B \subset \mathcal{N}$  that is not accepted and a set  $C$  in the adversary structure such that  $B \cup C = \mathcal{N}$ . For simplicity we prove the statement for the case where  $d = n$ , but the general proof follows from a very similar argument.

According to the properties of the ramp scheme there is an index  $i \in \{1, \dots, k\}$  such that  $\vec{\epsilon}_i$  is not in the image of  $M_B^T$ . Then  $\vec{\epsilon}_i \notin (\text{Ker}(M_B))^\perp$ , which means there must exist some element  $\vec{v} \in \text{Ker}(M_B)$  such that its  $i^{\text{th}}$  coordinate is nonzero.

Let  $\vec{v}' \in \mathbb{F}_q^k$  be the vector composed of the first  $k$  coordinates of  $\vec{v}$  and select a vector  $\vec{z}' \in \mathbb{F}_q^k$  such that  $\vec{v}' \odot \vec{z}' \neq \vec{0}$ , which exists since  $\odot$  is non-degenerate. Since  $C$  is a set in the adversary structure, there exists an element  $\vec{z}$  in the kernel of  $M_C$  such that its  $k$  first coordinates are those of  $\vec{z}'$ . Then we have  $M_B \vec{v} = \vec{0}$  and  $M_C \vec{z} = \vec{0}$ . Since the union of these sets covers  $\mathcal{N}$  we have that  $M \vec{v} * M \vec{z} = \vec{0}$ , where  $*$  denotes the component-wise product of the two vectors. Now, if the scheme were multiplicative, we would have that every coordinate of  $\vec{v}' \odot \vec{z}'$  is a linear combination of the coordinates of  $M \vec{v} * M \vec{z}$ . But we know that these coordinates are zero, and that  $\vec{v}' \odot \vec{z}' \neq \vec{0}$  so this gives a contradiction.  $\triangle$

PROPOSITION 5. *Suppose that there exists a  $\mathcal{A}$ -strongly multiplicative ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$ , with respect to some multiplication map  $\odot$ , that has access structure  $\Gamma$  and adversary structure  $\mathcal{A}$ . Then the tuple  $(\mathcal{A}, \Gamma)$  is  $R^{(3)}$ .*

PROOF. This follows from the definition of strong multiplication and Proposition 4.  $\triangle$

## Chapter 7

# Ramp Sharing Based on Random Error Correcting Codes

In this chapter we construct multiplicative ramp schemes over small fields and general ramp schemes with high information rate based on coding theory. In particular, the techniques in this chapter allow to construct families of multiplicative ramp schemes tolerating  $t$ -adversaries with  $t < (1/2 - \epsilon)n$  for any  $0 < \epsilon < 1/2$ , where the finite field can be kept constant while the number of participants increases. When used for the construction of multi-party computation protocols, this leads to passive-adversary multi-party computation protocols where the communication complexity no longer depends on the field size, resulting in a  $\log n$  efficiency increase compared to Shamir-based protocols.

Although the algebraic-geometric techniques used by Chen and Cramer [15] already achieve multiplicative schemes with similar properties, and in addition allow to construct families of strongly multiplicative ramp schemes tolerating  $t$ -adversaries with  $t < (1/3 - \epsilon)n$  for any  $0 < \epsilon < 1/3$ , our schemes are much easier to design and should be seen as complementary to their result. The novelty of this work lies in the introduction of an interesting new connection between ramp schemes on the one hand and error correcting codes with their dual code on the other hand. This connection allows us to introduce a new method of constructing multiplicative almost threshold ramp schemes from error correcting codes that is fundamentally different from the only previously known method of constructing such schemes, which is based on polynomial evaluation. In fact, our schemes can even be constructed from arbitrary (or even randomly chosen) error correcting codes.

Most of the material in this chapter can be found in [17].

## 7.1 Massey's Secret Sharing From Codes

Massey [56, 57] gave the following construction of a secret sharing scheme from an error correcting code. Let  $C$  be an  $[n + 1, k, d]$ -code over a finite field  $\mathbb{F}$ . The dual code  $C^\perp$  is then an  $[n + 1, n + 1 - k, d^\perp]$ -code. We assume in this section that  $C$  is non-degenerate, i.e., that the minimum distances of both  $C$  and  $C^\perp$  are greater than 1.

Let  $s \in \mathbb{F}$  be a secret value. Select a codeword  $\vec{c} = (c_0, c_1, \dots, c_n) \in C$  uniformly at random such that  $c_0 = s$ , and define the share-vector to be  $(c_1, \dots, c_n)$ . We denote this ramp scheme based on the code  $C$  by  $\text{RSS}(C)$ . The access structure  $\Gamma(C)$  is then determined as follows. For a vector  $\vec{x}$ , define

$$\text{sup}(\vec{x}) = \{i : x_i \neq 0\}$$

and let

$$C_0 = \{\vec{c} \in C : 0 \in \text{sup}(\vec{c})\}$$

be the subcode of  $C$  that is used for the ramp scheme  $\text{RSS}(C)$ . Then

$$\Gamma(C) = \{\text{sup}(\vec{c}^*) : \vec{c}^* \in (C_0)_{\{1,2,\dots,n\}}\},$$

where again  $C_A$  is used to denote the restriction of the code  $C$  to the positions in  $A$ .

It seems that Massey's primary interest was in demonstrating the correspondence between error correcting codes and secret sharing schemes. In particular, he does not consider the structure of the resulting access structures in his work. We now extend his idea in several ways in order to obtain suitable ramp schemes with almost-threshold access structures and prove bounds on their existence.

## 7.2 Extensions of Massey's Idea

We first state some known bounds on the parameters of the ramp scheme  $\text{RSS}(C)$  from Section 7.1 and provide a proof.

**THEOREM 23.** *Let  $C$  be an  $[n + 1, k]$ -code over a finite field  $\mathbb{F}$  and  $d = d_{\min}(C)$ . Then  $\text{RSS}(C)$  is a  $(d^\perp - 2, n - d + 2)$ -ramp scheme.*

**PROOF.** First, we argue that  $\Gamma(C) = (\Gamma(C^\perp))^*$ , i.e., the access structure of  $\text{RSS}(C)$  is the dual of the access structure of  $\text{RSS}(C^\perp)$ , and vice versa. Indeed,  $A \in \Gamma(C)$  if and only if there is  $\vec{c}^* \in C^\perp$  with  $(\vec{c}^*)_0 = 1$  and  $(\vec{c}^*)_i = 0$  for all  $i \in \{1, \dots, n\} \setminus A$  ( $:= \bar{A}$ ). The latter is a share vector with secret equal to 1 in

$\text{RSS}(C^\perp)$ , with shares equal to 0 for  $\bar{A}$ . The existence of such a share vector is equivalent to  $\bar{A} \notin \Gamma(C^\perp)$ . Now, from the characterization of  $\Gamma(C)$  it is immediate that  $\text{RSS}(C)$  rejects all sets of size  $d^\perp - 2$ . Since  $\text{RSS}(C^\perp)$  rejects all sets of size  $d - 2$  and since  $\Gamma(C) = (\Gamma(C^\perp))^*$ , it must be that  $\text{RSS}(C)$  accepts all sets of size  $n - d + 2$ .  $\triangle$

The exact rejection threshold  $t_{\max}$  is equal to

$$-2 + \min\{w_H(\vec{c}^*) : \vec{c}^* \in C^\perp : (\vec{c}^*)_0 = 1\},$$

i.e., this is the largest cardinality such that the joint shares of any set of this cardinality give no information on the secret. The exact acceptance threshold  $r_{\min}$  is equal to

$$n + 2 - \min\{w_H(\vec{c}) : \vec{c} \in C : \vec{c}_0 = 1\}.$$

For  $A \subset \{1, \dots, n\}$ , let  $\phi_A(C)$  denote the code restricted to the coordinates from the set  $i \in A \cup \{0\}$ , i.e., consisting of all codewords of  $C$  stripped of the coordinates not in  $A \cup \{0\}$ .

**DEFINITION 40.** *A code  $C$  is weakly self-dual if there is a diagonal matrix  $W \in \mathbb{F}^{n+1, n+1}$  such that  $w_{00} = 1$  and  $W\vec{c} \in C^\perp$  for all  $\vec{c} \in C$ . A code  $C$  is  $t$ -locally weakly self-dual if for all sets  $B \subset \{1, \dots, n\}$  with  $|B| = n - t$  the code  $\phi_B(C)$  is weakly self-dual.*

Recall that a code  $C$  is self-dual if  $C = C^\perp$ . We note that our definition for weakly self-dual codes is a slight relaxation of the notion of quasi self-orthogonal<sup>1</sup> codes, while the  $t$ -local variation appears to be novel. Simple examples are the following: the  $[n + 1, t + 1, n - t + 1]$ -Reed Solomon code is weakly self-dual if  $t < \frac{n}{2}$  and  $t$ -locally weakly self-dual if  $t < \frac{n}{3}$ . The following theorem demonstrates the relevance of these notions in secure computation.

**THEOREM 24.** *If  $C$  is a self-dual code of length  $n + 1$  with minimum distance  $d$ , then  $\text{RSS}(C)$  is a  $(t, n - t)$ -multiplicative ramp scheme. If  $C$  is weakly self-dual, then  $\text{RSS}(C)$  is multiplicative and  $t = d^\perp - 2$  if the matrix  $W$  is regular and otherwise  $t = \min\{d - 2, d^\perp - 2\}$ . If  $C$  is  $t$ -locally weakly self-dual then  $\text{RSS}(C)$  is  $\mathcal{A}$ -strongly multiplicative with respect to the  $t$ -adversary structure  $\mathcal{A}$ .*

**PROOF.** Since  $d = d^\perp$  for self-dual codes, the rejection and acceptance claims follow from Theorem 23. From  $\langle \vec{c}, \vec{c}' \rangle = 0$  for all  $\vec{c}, \vec{c}' \in C$  we get

$$c_0 c'_0 = -c_1 c'_1 - \dots - c_n c'_n.$$

This implies the resulting scheme is multiplicative. For weakly self-dual codes, if  $W$  is regular then the minimum distance of  $WC$  is the same as that of  $C$ . Since  $WC \subset C^\perp$ ,

<sup>1</sup>For quasi self-orthogonal codes, the matrix  $W$  is required to be regular.

we must have  $d^\perp \leq d$ , and we apply Theorem 23. As to multiplication, we now have  $\langle W\vec{c}, \vec{c}' \rangle = 0$ , so

$$c_0 c'_0 = -w_1 c_1 c'_1 - \cdots - w_n c_n c'_n.$$

The claim about the strong multiplication property is now obvious from the definition.  $\triangle$

We can generalize this as follows, using a twist on an idea from Cramer, Damgård and Maurer [23]. Let  $C$  be a code of length  $n + 1$  and minimum distance  $d$ . Consider the linear secret sharing scheme  $\text{RSS}^\dagger(C)$  defined as follows. Take the secret  $s$ , and generate random shares  $(c_1, \dots, c_n)$  according to  $\text{RSS}(C)$ , and generate independently random shares  $(c_1^*, \dots, c_n^*)$  according to  $\text{RSS}(C^\perp)$ . The share vector is then defined as  $((c_1, c_1^*), \dots, (c_n, c_n^*))$ .

**THEOREM 25.** *Let  $C$  be a code of length  $n + 1$  and minimum distance  $d$ . Define  $t(C) = \min\{d - 2, d^\perp - 2\}$ . Then  $\text{RSS}^\dagger(C)$  is a  $(t(C), n - t(C))$ -multiplicative ramp scheme. In particular,  $t(C) < n/2$ .*

The claim that  $t(C) < n/2$  can for instance be verified by applying the Singleton-bound to  $C$  as well as to  $C^\perp$ . Note however that for this scheme the shares are twice as large as the secret.

Strong multiplication is much more elusive and is not achieved by the construction above. In fact, the only way known to ensure strong multiplication is using constructions based on algebraic-geometric codes or their classical special cases with genus  $g = 0$ , which can all be found in Chapter 8.

### 7.3 Existence and Bounds

Our main objective in this section is to prove several lower bounds on the maximal value  $T$  taken over all values  $t = \min\{d - 2, d^\perp - 2\}$  as  $C$  ranges over all  $\mathbb{F}$ -linear codes of length  $n + 1$  and minimum distance  $d$ . The bounds we achieve are of independent interest due to the fact that they allow to attach an error probability to estimates on the parameters of randomly selected codes.

We then use these bounds to prove the existence of families of multiplicative ramp schemes tolerating  $t$ -adversaries with  $t < (1/2 - \epsilon)n$  for any  $0 < \epsilon < 1/2$ , where the finite field can be kept constant while the number of participants increases. Using the error probabilities attached to our bounds, we simultaneously demonstrate that the schemes in these families can be obtained with arbitrarily high probability by selecting random codes with appropriate dimension and code length.

### 7.3.1 General lower bounds on $T$

In Theorem 27 we give a general lower bound on the maximal  $t$ . In Corollary 7 we treat the general case when  $\mathbb{F} = \mathbb{F}_2$ . In Theorem 28 we show that one can asymptotically get arbitrarily close to  $n/2$ , over some constant size field. More generally, we treat in the same theorem the parametrized case where  $C$  is randomly selected and a security parameter regulates the error probability that  $t$  is below a certain bound.

In the following, we let  $q$  be some fixed prime power.

DEFINITION 41. *Let  $n \in \mathbb{Z}_{>0}$  be fixed. Then  $T(n+1, q) := \max_C t(C)$ , where  $C$  ranges over all subcodes of  $\mathbb{F}_q^{n+1}$ . Similarly,  $T'(n+1, q) := \max_C t(C)$ , where  $C$  ranges over all weakly self-dual subcodes of  $\mathbb{F}_q^{n+1}$ .*

DEFINITION 42. *Let  $\mathcal{C}_k$  have the uniform distribution over the set of  $[n+1, k]$ -subcodes of  $\mathbb{F}_q^{n+1}$ . Then we define*

$$T(n+1, q, m, k) := \max\{d-2 : P(\min\{d_{\min}(\mathcal{C}_k), d_{\min}(\mathcal{C}_k^\perp)\} < d) < 2^{-m}\}$$

and  $T(n+1, q, m) := \max_k T(n+1, q, m, k)$ .

It is easy to see that  $T(n+1, q) \geq T(n+1, q, 0)$ . The following lemma is trivial.

LEMMA 9. *Suppose  $k \leq n$ . For each pair  $(\vec{x}, \vec{y})$  with  $\vec{x} \in \mathbb{F}_q^k \setminus \{0\}$  and  $\vec{y} \in \mathbb{F}_q^n \setminus \{0\}$  there exists an  $n \times k$  matrix  $M$  of rank  $k$  such that  $M\vec{x} = \vec{y}$ .*

The following theorem bounds the probability that a randomly chosen code has a minimum distance less than some fixed value  $d$ . It is used for most of the bounds that follow later.

THEOREM 26. *Let  $\mathcal{C}$  have the uniform distribution over the set of  $[n, k]$ -subcodes of  $\mathbb{F}_q^n$ . Furthermore assume that  $d = \alpha n \in \mathbb{Z}$ , where  $0 < \alpha < \frac{1}{2}$ . Then*

$$P(\exists \vec{y} \in \mathcal{C} \setminus \{0\} : w_H(\vec{y}) < d) < q^{k+n(H_q(\alpha)-1)}.$$

PROOF. Let  $\mathcal{H}$  have the uniform distribution over the set of  $n \times k$  matrices of rank  $k$  over  $\mathbb{F}_q$ . Every such matrix corresponds to an ordered basis for a subcode  $V$  of  $\mathbb{F}_q^n$ . Since there is a one-to-one correspondence between the ordered bases for  $V$  and the linear isomorphisms between  $V$  and  $\mathbb{F}_q^k$ , each such subcode has the same number of ordered bases. Therefore, the variable  $\mathcal{H}$  induces a uniformly random selection of an  $[n, k]$ -subcode of  $\mathbb{F}_q^n$ .

Fix some non-zero  $\vec{x} \in \mathbb{F}_q^k$ . The variable  $\mathcal{H}\vec{x}$  then corresponds to a uniformly random selection from  $\mathbb{F}_q^n$ , which can be seen as follows: First, by Lemma 9 for any non-zero  $\vec{y} \in \mathbb{F}_q^n$  there exists an  $n \times k$  matrix  $M$  of rank  $k$  such that  $M\vec{x} = \vec{y}$ . Now fix some  $\vec{y} \in \mathbb{F}_q^n$  and assume that  $M\vec{x} = \vec{y}$  for some  $n \times k$ -matrix  $M$  of rank  $k$ . Then

$\#\{M' : M'\vec{x} = \vec{y}\} = \#\{M' : (M - M')\vec{x} = 0\} = \#\{M' : M'\vec{x} = 0\}$ , so for every  $\vec{y} \in \mathbb{F}_q^n$  there are the same number of matrices of rank  $k$  such that  $M\vec{x} = \vec{y}$ .

Now let  $\vec{x}$  range over the elements of  $\mathbb{F}_q^k$ . It follows that

$$\begin{aligned}
 P(\exists \vec{y} \in \mathcal{C} \setminus \{0\} : w_H(\vec{y}) < d) &= P(\exists \vec{x} \in (\mathbb{F}_q^k)^* : w_H(\mathcal{H}\vec{x}) < d) \\
 &\leq \sum_{\vec{x} \in (\mathbb{F}_q^k)^*} P(w_H(\mathcal{H}\vec{x}) < d) \\
 &= \frac{q^k - 1}{q^n - 1} \cdot \sum_{i=1}^{d-1} \binom{n}{i} (q-1)^i \\
 &< \frac{q^k}{q^n} \cdot (q-1)^d \sum_{i=1}^{d-1} \binom{n}{i} \\
 &< \frac{q^k}{q^n} \cdot q^{\alpha n \log_q(q-1)} \cdot 2^{nH_2(\alpha)} \\
 &= q^{k+n(H_q(\alpha)-1)},
 \end{aligned}$$

where the last inequality is a consequence of Lemma 2.  $\triangle$

Since there is a one-to-one correspondence between linear subcodes  $C \subset \mathbb{F}_q^n$  and their dual codes  $C^\perp$ , the random variable  $\mathcal{C}^\perp$  corresponds to a uniformly random selection from the set of  $[n, n-k]$ -subcodes of  $\mathbb{F}_q^n$ . Therefore, we immediately obtain the following corollary.

**COROLLARY 6.** *Let  $\mathcal{C}$  have the uniform distribution on the set of  $[n, k]$ -subcodes of  $\mathbb{F}_q^n$ . Furthermore assume that  $d^* = \alpha n \in \mathbb{Z}$ , where  $0 < \alpha < \frac{1}{2}$ . Then*

$$P(\exists \vec{y} \in \mathcal{C}^\perp \setminus \{0\} : w_H(\vec{y}) < d^*) < q^{nH_q(\alpha)-k}.$$

Using the fact that  $-\lambda \ln \lambda - (1-\lambda) \ln(1-\lambda) < 3.3\lambda$  for  $1/10 \leq \lambda \leq 1/2$ , we obtain that

$$H_q(\lambda) < \lambda \log_q(q-1) - \frac{3.3}{\ln q} \lambda \quad (7.1)$$

for  $1/10 \leq \lambda \leq 1/2$ . This gives rise to the following theorem that gives a general lower bound on  $t$  for codes in the space  $\mathbb{F}_q^{n+1}$  that is correct with probability at least  $1 - 2^{-m}$  when such a code is selected at random.

**THEOREM 27.**  $T(n+1, q, m) \geq \lfloor \beta(n+1, q, m) \rfloor - 2$  with

$$\beta(n+1, q, m) = \frac{(n+1) \ln q - 2(m+1) \ln 2}{2 \ln(q-1) + 6.6},$$

provided that  $\lfloor \beta(n+1, q, m) \rfloor \geq n/10$ .

### 7.3. Existence and Bounds

---

PROOF. Set  $k = (n + 1)/2$  and let  $\mathcal{C}$  be as in Theorem 26. By Theorem 26 and Corollary 6,

$$\begin{aligned} P(\min\{d_{\min}(\mathcal{C}), d_{\min}(\mathcal{C}^\perp)\} < d) &\leq P(d_{\min}(\mathcal{C}) < d) + P(d_{\min}(\mathcal{C}^\perp) < d) \\ &< 2 \cdot q^{(n+1)H_q(\alpha) - (n+1)/2}. \end{aligned}$$

We want  $P(\min\{d_{\min}(\mathcal{C}), d_{\min}(\mathcal{C}^\perp)\} < d) < 2^{-m}$ . Filling in (7.1) and rewriting, we see that this is the case if

$$d \leq \frac{(n + 1) \ln q - 2(m + 1) \ln 2}{2 \ln(q - 1) + 6.6}.$$

△

The following corollary gives a rough estimate for the case  $\mathbb{F} = \mathbb{F}_2$ . A tighter estimate can be found in Corollary 8.

COROLLARY 7. *If  $n \geq 21$ , then  $T(n + 1, 2) \geq \lfloor 0.1n \rfloor - 2$ .*

The theorem below can be considered complementary to the result of Chen and Cramer [15] and together with our construction of ramp schemes from codes demonstrates that one can achieve infinite families of multiplicative ramp schemes with near-optimal parameters over constant-size fields. Furthermore, the theorem demonstrates that one can select suitable codes for such constructions with arbitrarily small error probability given a large enough code length  $n + 1$ .

THEOREM 28. *Fix any arbitrarily small  $\epsilon > 0$  and any  $m \in \mathbb{Z}_{>0}$ . Then there exists a fixed finite field  $\mathbb{F}_q$  over which for infinitely many  $n$  there exist  $[n + 1, k]$ -codes  $C \subset \mathbb{F}_q^{n+1}$  with  $(1/2 - \epsilon)n \leq t(C) \leq n/2$  where such a code can be selected with probability at least  $1 - 2^{-m}$  using a random selection among the  $[n, k]$ -subcodes of  $\mathbb{F}_q^{n+1}$ .*

PROOF. Let  $d$  be the minimum distance of  $C$  and  $d^\perp$  the minimum distance of  $C^\perp$ . By Theorem 25,  $t(C) < n/2$ . Therefore, it suffices to show that  $(d - 2)$  and  $(d^\perp - 2)$  can simultaneously get arbitrarily close to  $n/2$  (relative to  $n$ ) with probability at least  $1 - 2^{-m}$ .

By Theorem 27,

$$T(n + 1, q, m) \geq \beta(n + 1, q, m) - 2 = \frac{(n + 1) \ln q - 2(m + 1) \ln 2}{2 \ln(q - 1) + 6.6} - 2$$

and we have that

$$\begin{aligned} \lim_{q \rightarrow \infty} \frac{(n + 1) \ln q - 2(m + 1) \ln 2}{2 \ln(q - 1) + 6.6} - 2 &= \lim_{q \rightarrow \infty} \frac{(n + 1) \ln q}{2 \ln(q - 1) + 6.6} - 2 \\ &\geq \lim_{q \rightarrow \infty} \frac{(n + 1) \ln q}{2 \ln q + 6.6} - 2. \end{aligned}$$

Since  $\lim_{x \rightarrow \infty} \frac{x}{x+3.3} = \lim_{y \rightarrow \infty} \frac{y-3.3}{y} = \lim_{y \rightarrow \infty} (1 - \frac{3.3}{y}) = 1$ , the final term converges to  $(n+1)/2 - 2$  as  $q \rightarrow \infty$ . We can therefore for any  $\delta > 0$  select a  $q$  large enough such that  $T(n, q, m) \geq n/2 - 3/2 - \delta$ . For large enough  $n$ ,  $(3/2 + \delta)/n < \epsilon$  and the claim follows.  $\triangle$

So far we have assumed a random selection from the set of  $[n, k]$ -subcodes of  $\mathbb{F}_q^n$ . The lemma below demonstrates, together with the proof of Theorem 26, that we can in fact perform this random selection by selecting  $n \times k$  matrices at random, where we obtain a matrix of rank  $k$  with probability at least  $1/4$ .

LEMMA 10. *The probability that a randomly selected  $n \times k$ -matrix over  $\mathbb{F}_q$  has full rank is larger than  $1 - 1/q - 1/q^2$ .*

*Proof.* An  $i$ -dimensional space over  $\mathbb{F}_q$  contains  $1 + (q-1)^i \leq q^i$  points. Therefore, the probability that a randomly selected codeword falls outside of an  $i$ -dimensional subspace is at least  $1 - q^{i-n}$ . It follows that a randomly selected  $(n \times k)$ -matrix has full rank with probability at least  $\prod_{i=0}^{k-1} (1 - q^{i-n})$ , with the worst case occurring when  $k = n$ .

We now prove by induction that

$$\prod_{i=0}^{n-1} (1 - q^{i-n}) = \prod_{i=1}^n (1 - q^{-i}) \geq \left(1 - \frac{1}{q} - \frac{1}{q^2}\right) + \frac{1}{q^{n+1}},$$

from which the claim follows. First note that  $\prod_{i=1}^1 (1 - q^{-1}) = 1 - 1/q = 1 - 1/q - 1/q^2 + 1/q^2$ , so the base case is correct. Now assume that  $\prod_{i=1}^j (1 - q^{-i}) \geq 1 - 1/q - 1/q^2 + 1/q^{j+1}$  for  $j = 1, 2, \dots, m-1$ . Then

$$\begin{aligned} \prod_{i=1}^m (1 - q^{-i}) &\geq \left(1 - \frac{1}{q} - \frac{1}{q^2} + \frac{1}{q^m}\right) \left(1 - \frac{1}{q^m}\right) \\ &= \left(1 - \frac{1}{q} - \frac{1}{q^2}\right) + q \cdot \frac{1}{q^{m+1}} - \left(q - 1 - \frac{1}{q}\right) \cdot \frac{1}{q^{m+1}} - \\ &\quad \frac{1}{q^{m-1}} \cdot \frac{1}{q^{m+1}} \\ &> \left(1 - \frac{1}{q} - \frac{1}{q^2}\right) + \frac{1}{q^{m+1}} \end{aligned}$$

□

### 7.3.2 Bounds from (Weakly) Self-Dual Codes

In Corollary 8 we prove a general lower bound on  $T$  for binary self-dual codes, and Theorem 30 shows that for  $n < 100$  the situation is much better than the bound indicates. We are especially interested in self-dual codes, because secret sharing schemes

### 7.3. Existence and Bounds

---

based on self-dual codes do not suffer from the  $1/2$  information rate loss that occurs in the general case. Finally, in Theorem 31 we prove a much better lower bound for weakly self-dual codes based on algebraic geometry, and not random codes. Note that the results based on algebraic geometry are only known to hold if the size of the field is a square.

**THEOREM 29.** *Let  $n$  be any positive integer and let  $d_{GV}$  be the largest integer such that*

$$\sum_{\substack{0 < i < d \\ 2|i}} \binom{n}{i} < 2^{n/2-1} + 1.$$

*Then there exists a self-dual binary code of length  $n$  and minimum distance at least  $d_{GV}$ .*

**PROOF.** See [54, 74, 61]. △

**COROLLARY 8.** *Fix  $\epsilon > 0$ . For large enough  $n$ ,  $T'(n, 2) \geq \lfloor (\delta - \epsilon)n \rfloor - 2$ , where  $\delta \approx 0.11002786$  is any truncated approximation of the unique solution less than  $1/2$  of  $H_2(\delta) = 1/2$ .*

**PROOF.** ([54, 74, 61]) Let  $d = \alpha(n + 1)$ . Since for  $\alpha < 1/2$ ,  $\sum_{0 < i < d} \binom{n+1}{i} \leq 2^{(n+1)H(\alpha)}$ , the conditions of Theorem 29 are met if

$$(n + 1)H(\alpha) \leq \frac{n + 1}{2} - 1 \Leftrightarrow H(\alpha) \leq \frac{1}{2} - \frac{1}{n + 1}.$$

The solution for  $\alpha$  then comes arbitrarily close to  $\delta$  as  $n$  increases. △

**THEOREM 30.** *There exist self-dual binary codes  $C$  of length  $n + 1 < 100$  for which  $d_{\min}(C) > n/5$ . In particular, there exist self-dual binary codes  $C$  with the following parameters:*

| $n + 1$ | $d_{\min}(C)$ |
|---------|---------------|
| 12      | 4             |
| 22      | 6             |
| 24      | 8             |
| 46      | 10            |
| 48      | 12            |

**PROOF.** See [35]. △

**THEOREM 31.** *When we take the maximum over algebraic-geometric codes, then*

$$T(n + 1, q^2) > \left( \frac{1}{2} - \frac{1}{q - 1} \right) n.$$

PROOF. This follows from a suitable choice of parameters for algebraic-geometric codes and their duals and the existence of Garcia-Stichtenoth curves, using techniques similar to those in [15].  $\triangle$

## 7.4 High Information Rate Ramp Schemes

We generalize Massey's scheme from Section 7.1 to create high information rate ramp schemes in Section 7.4.1. In Section 7.4.2, we give a completely general construction that does not consume code length (which corresponds to the number of players in the scheme) for an increased information rate. As an application we use this theory to analyze an alternative high information rate ramp scheme based on Shamir. Also, our general method gives rise to a new high information rate ramp scheme based on algebraic-geometric code which we introduce in Section 7.4.3.

### 7.4.1 A High Information Rate Ramp Scheme

Let  $C$  be an  $[n + \ell, k]$ -code over a finite field  $\mathbb{F}$  with minimum distance  $d$ . We now extend Massey's scheme from Section 7.1 in the direction of high information rate as follows. Let  $\ell$  be a non-negative integer such that  $\ell < d^\perp$ .

Let  $\vec{s} \in \mathbb{F}^\ell$ . Select a codeword  $\vec{c} = (c'_0, \dots, c'_{\ell-1}, c_1, \dots, c_n) \in C$  at random such that  $\vec{s} = (c'_0, \dots, c'_{\ell-1})$ . Such  $\vec{c}$  always exists. Define the coefficients of  $(c_1, \dots, c_n)$  to be the shares. We claim that this is a  $(d^\perp - \ell - 1, n + \ell - d + 1)$ -ramp scheme with secrets of length  $\ell$ . This can be verified from the following facts.

Acceptance follows from the fact that if there would exist two codewords in  $C$  that agreed on  $n + \ell - d + 1$  share locations, their difference would give a codeword in  $C$  with Hamming weight less than  $d$ . As for rejection, note that in a generator matrix for  $C$ , any collection of  $m < d^\perp$  rows (the code is generated by the columns) are linearly independent. So the corresponding columns span  $\mathbb{F}^m$ . Therefore, for each  $j \in \{0, \dots, \ell - 1\}$  and for each  $A \subset \{1, \dots, n\}$  with  $|A| \leq d^\perp - \ell - 1$  there exists a codeword  $\vec{c}$  such that  $c'_j = 1$  and  $c'_i = 0$  for all  $i \in \{0, \dots, \ell - 1\} \setminus \{j\}$  and  $c_u = 0$  for all  $u \in A$ . This implies rejection as claimed.

### 7.4.2 A More Fruitful Approach

A disadvantage of the scheme above is that it consumes code-length in exchange for secret-length. Below we describe an entirely general approach that doesn't have this disadvantage, and by means of which one can prove the existence of improved ramp schemes (see Section 7.4.3).

Let  $\hat{C}$  and  $C$  be linear codes of length  $n$  over  $\mathbb{F}$ , i.e., they are subspaces of the vector space  $\mathbb{F}^n$ . Assume that  $C$  has dimension greater than 0 and that it is a proper

subspace of  $\hat{C}$ . Choose an arbitrary linear code  $S$  such that

$$\hat{C} = S + C \text{ and } S \cap C = \{0\},$$

i.e., a direct sum. This is always possible of course, for instance by completing a basis of  $C$  to one of  $\hat{C}$ . Write

$$\ell = \dim_{\mathbb{F}}(\hat{C}) - \dim_{\mathbb{F}}(C) (= \dim_{\mathbb{F}}(S))$$

and fix an arbitrary isomorphism  $\psi : \mathbb{F}^{\ell} \rightarrow S$ .

We now define the following linear ramp scheme. Let  $\vec{s} \in \mathbb{F}^{\ell}$  be the secret vector. Sample uniformly at random  $\vec{c} \in C$  and define the share vector  $\vec{\hat{c}}$  as  $\vec{\hat{c}} = \psi(\vec{s}) + \vec{c}$ .<sup>2</sup>

Note that this is a generalization of a scheme used by Ozarow and Wyner [59], who considered the case  $\hat{C} = \mathbb{F}^n$ . In fact, all possible linear ramp schemes are captured by this general scheme we consider here.

For  $A \subset \{1, \dots, n\}$ , let  $\phi_A$  denote the function

$$\phi_A : \mathbb{F}^n \rightarrow \mathbb{F}^{|A|}$$

where

$$(x_1, \dots, x_n) \mapsto (x_i)_{i \in A},$$

i.e., restriction to the coordinates labeled with  $A$ . Given  $A$ , consider the restriction of  $\phi_A$  to  $\hat{C}$ . The set  $A$  is said to be rejected if the collection of shares  $\{\hat{c}_i\}_{i \in A}$  give no information on the secret vector, and accepted if those shares always determine the secret vector uniquely.

**THEOREM 32.** *Let  $\ell = \dim(\hat{C}) - \dim(C)$ . The set  $A$  is rejected if and only if  $\dim(\phi_A(\hat{C})) - \dim(\phi_A(C)) = 0$ . The set  $A$  is accepted if and only if  $\dim(\phi_A(\hat{C})) - \dim(\phi_A(C)) = \ell$ . More generally, the uncertainty about the secret vector  $\vec{s}$ , given the shares of  $A$ , is equal to  $r$  elements of  $\mathbb{F}$ , where  $r$  is such that  $\ell - r = \dim(\phi_A(\hat{C})) - \dim(\phi_A(C))$ .*

**PROOF.** Rejection (for the set  $A$ ) is equivalent to saying that for each possible secret vector  $\vec{s} \in \mathbb{F}^{\ell}$ , there is a share vector  $\vec{\hat{c}}$  that “encodes”  $\vec{s}$  and that satisfies  $\phi_A(\vec{\hat{c}}) = 0$ . This is the same as saying that for each  $\vec{z} \in S$ , there exists  $\vec{c} \in C$  such that  $0 = \phi_A(\vec{z} + \vec{c}) = \phi_A(\vec{z}) + \phi_A(\vec{c})$ . Thus,  $\phi_A(\hat{C}) \subset \phi_A(C)$ . Since the other inclusion holds regardless of  $A$ , the rejection claim follows.

As for acceptance (for the set  $A$ ), this is equivalent to saying that there are no two distinct  $\vec{z}, \vec{z}' \in S$  so that  $\phi_A(\vec{z} + \vec{c}) = \phi_A(\vec{z}' + \vec{c})$  for some  $\vec{c}, \vec{c}' \in C$ . This is equivalent

---

<sup>2</sup>Equivalently, one can say that we fixed an arbitrary isomorphism from  $\mathbb{F}^{\ell}$  to  $\hat{C}/C$ , and that the share vector is selected by mapping  $\vec{s}$  to the residue-class of  $\psi(\vec{s})$  modulo  $C$ , and that  $\vec{\hat{c}}$  is chosen uniformly at random from that residue-class.

to saying that  $\dim(\phi_A(S)) = \ell$  and  $\phi_A(S) \cap \phi_A(C) = \{0\}$ . Since  $\dim(\phi_A(\hat{C})) - \dim(\phi_A(C)) = \dim(\phi_A(S)) - \dim(\phi_A(S) \cap \phi_A(C))$ , the acceptance claim follows. The cases in between these two extremes should now be obvious.  $\triangle$

We give the following estimate with respect to rejection and acceptance (which, as one can prove by giving counter-examples, is not always sharp).

**COROLLARY 9.** *The set  $A$  is rejected if  $|A| < d_{\min}(C^\perp)$ . The set  $A$  is accepted if  $|A| > n - d_{\min}(\hat{C})$ .*

**PROOF.** As for rejection, if  $|A| < d_{\min}(C^\perp)$ , then  $\phi_A(C)$  clearly has rank  $|A|$ , since otherwise we could construct a codeword in  $C^\perp$  whose weight is smaller than  $d_{\min}(C^\perp)$ . Since  $\phi_A(C) \subset \phi_A(\hat{C}) \subset \mathbb{F}^{|A|}$ , we must have  $\phi_A(C) = \phi_A(\hat{C})$ , and rejection follows from the theorem.

As for acceptance, if  $|A| > n - d_{\min}(C)$ , then  $\phi_A(\vec{c}) = \vec{0}$  if and only if  $\vec{c} = \vec{0}$ , since otherwise  $C$  would contain a codeword whose weight is smaller than  $d_{\min}(C)$ . Thus,  $\phi_A$  is injective when restricted to  $\hat{C}$ , and  $\vec{c}$  follows uniquely from  $\phi_A(\vec{c})$ . Since  $S \cap C = \{0\}$ ,  $\psi(\vec{s})$  and  $\vec{c}$  follow uniquely from  $\vec{c}$ . The secret vector  $\vec{s}$  now follows uniquely from  $\psi(\vec{s})$  since  $\psi$  is bijective.  $\triangle$

Note that from the Singleton-bound, we have  $\dim_{\mathbb{F}}(\hat{C}) \leq n - d_{\min}(\hat{C}) + 1$  and  $d_{\min}(C^\perp) - 1 \leq n - \dim_{\mathbb{F}}(C^\perp) = \dim_{\mathbb{F}}(C)$ . Thus,  $r - t \geq \dim_{\mathbb{F}}(\hat{C}) - \dim_{\mathbb{F}}(C)$  in any linear ramp scheme.

Before presenting constructive results, we as an example analyze a Shamir-type ramp scheme with this theory. Suppose  $n > |\mathbb{F}|$ , and let  $x_1, \dots, x_n$  be distinct non-zero elements of  $\mathbb{F}$ . Consider the Vandermonde matrix  $M$  with  $n$  rows and  $t$  columns whose  $i$ -th row is  $(1, x_i, \dots, x_i^t)$ . Let  $\hat{C}$  be the code generated by all the columns. This is an  $(n, t+1, n-t)$ -MDS code. So its dual is an  $(n, n-t-1, t+2)$ -code. Let  $C$  be the code generated by the last  $t+1-\ell$  columns. Clearly  $C \subset \hat{C}$ . By appropriately scaling the rows of  $C$  it is immediate that  $C$  is equivalent to an  $(n, t+1-\ell, n-t+\ell)$ -code. This is an MDS code, so its dual is an  $(n, n-t-1+\ell, t+2-\ell)$ -code. So by our theorem the resulting ramp scheme rejects all sets of size  $t+1-\ell$ , and accepts all sets of size  $t+1$ . Note that the gap between the two bounds here is  $\ell$ , so that is optimal.

### 7.4.3 High Information Rate Ramp Schemes: Existence and Bounds

In this section we demonstrate two methods for constructing high information rate ramp schemes. First, we present a new high information rate ramp scheme that improves the one presented in [15], where  $\hat{C}$  will be an algebraic-geometric code and  $C$

will be a carefully selected algebraic-geometric subcode of  $\hat{C}$ . Then, we demonstrate that high information rate ramp schemes can be obtained from random codes and bound the error probabilities on their predicted parameters.

### Algebraic-Geometric Codes

Select an absolutely irreducible smooth projective curve over a finite field  $\mathbb{F}$ , write  $g$  for its genus and let  $\{Q, P_1, P_2, \dots, P_n\}$  denote distinct points on the curve. Consider the rational divisor  $\hat{D} = (2g+t) \cdot Q$ , and let  $L(\hat{D})$  denote the corresponding Riemann-Roch space of  $\mathbb{F}_q$ -rational functions. Write  $\hat{C}$  for the Goppa-code consisting of the codewords  $(f(P_1), \dots, f(P_n))$ , where  $f$  ranges over  $L(\hat{D})$ . Also define the  $\mathbb{F}_q$ -rational divisor  $D = (2g+t-\ell) \cdot Q$ , and let  $L(D)$  denote the corresponding Riemann-Roch space of rational functions. Write  $C$  for the Goppa-code consisting of the codewords  $(f(P_1), \dots, f(P_n))$ , where  $f$  ranges over  $L(D)$ .

By the Riemann-Roch Theorem the dimension of  $\hat{C}$  is  $g+t+1$ , whereas the dimension of  $C$  is  $g+t+1-\ell$ . Since  $\hat{D} \geq D$ , we have  $L(D) \subset L(\hat{D})$ , and hence  $C \subset \hat{C}$ . It is fact that the minimum distance of  $C^\perp$  is at least  $\deg(D)-2g+2 = t-\ell+2$ . Furthermore, we show in Chapter 8 that we have acceptance for  $\deg(\hat{D})+1 = 2g+t+1$  shares. Thus, by our theorem, we have a  $(t-\ell+1, 2g+t+1)$ -ramp scheme with secrets of length  $\ell$ . Note that the improvement consists in the fact that the scheme above does not use up any points on the curve in order to encode the secret vector. Also note that by taking the projective line (i.e.,  $g=0$ ) we recover the earlier Shamir ramp scheme example. Using Garcia-Stichtenoth towers [36] our ramp scheme can be defined over constant size fields. See Chapter 8 for more details.

### Random Codes

Finally, the results in Section 7.3 demonstrate that we can also obtain high information rate ramp schemes from randomly selected codes  $\hat{C}$  and  $C$ , provided that  $C \subset \hat{C}$ . Theorem 32 demonstrates that for such codes  $C$  and  $\hat{C}$ , the corresponding ramp scheme reject any subset consisting of at most  $d_{\min}(C^\perp) - 1$  players and accept any subset consisting of at least  $n - d_{\min}(\hat{C}) + 1$  players.

One method of obtaining the appropriate distribution for  $C$  and  $\hat{C}$ , as demonstrated in the proof of Theorem 26, is to randomly select a matrix  $M$  from the set of  $n \times \hat{k}$ -matrices of rank  $\hat{k}$  and let  $\hat{C}$  be the code spanned by the columns. It is easy to see that if we now look at the last  $k$  columns of  $M$ , these columns in turn span a random  $[n, k]$ -subcode  $C$  of  $\mathbb{F}^n$  that is furthermore contained in  $\hat{C}$ . Clearly, the corresponding scheme allows for a secret vector of length  $\ell = \hat{k} - k$ .

Suppose that we want the scheme to reject all set of size at most  $t$  players and accept all sets of size at least  $n - \hat{t}$ . Using a similar argument as in Theorem 26 and using the fact that  $-\lambda \ln \lambda - (1-\lambda) \ln(1-\lambda) < 1.2\sqrt{\lambda}$  for  $0 \leq \lambda \leq 1/2$ , the following

theorem is now straightforward to obtain. It provides, for many different parameters and with arbitrarily high probability, a lower bound on  $t$  and  $\hat{t}$  when we select the codes  $C$  and  $\hat{C}$  at random.

**THEOREM 33.** *Select an  $[n, k]$ -code  $C$  and an  $[n, \hat{k}]$ -code  $\hat{C}$  over  $\mathbb{F}_q$  at random under the restriction that  $C \subset \hat{C}$ . Then*

$$P(d_{\min}(C^\perp) < t) < q^{-(k-t \log_q(q-1) - \frac{1.2\sqrt{tn}}{\ln q})}$$

and

$$P(d_{\min}(\hat{C}) < \hat{t}) < q^{-(n-\hat{k}-\hat{t} \log_q(q-1) - \frac{1.2\sqrt{\hat{t}n}}{\ln q})}.$$

## Chapter 8

# Ramp Sharing Based on Algebraic Geometry

In this chapter we describe all general constructions for almost-threshold strongly multiplicative ramp schemes known in the literature. These schemes have tight parameters, in the sense that they can tolerate  $t$ -adversaries for any  $t < (1/3 - \epsilon)n$ , where  $0 < \epsilon < 1/3$  can be made arbitrarily small.

First we describe the strongly multiplicative ramp scheme discovered by Franklin and Yung [34]. We then describe our new strongly multiplicative ramp scheme (Cramer, Damgård and de Haan [21]) followed by our later discovered improved version of this scheme (Chen, Cramer, de Haan and Cascudo [16]). These schemes all have high information rate.

We then proceed with a description of the algebraic-geometric strongly multiplicative ramp schemes of Chen and Cramer [15] and Chen, Cramer, de Haan and Cascudo [16] that can be seen as generalizations of the initial schemes in this chapter. In particular, the algebraic-geometric schemes offer both a high information rate and small, or even constant, field sizes, thus resolving both limitations from Chapter 6 at the same time.

### 8.1 Classical Ramp Schemes

We are now ready to introduce some classical high information rate ramp schemes. The ramp schemes in this section can all be seen as extensions of Shamir's secret sharing scheme that are defined for one of the following two multiplication maps  $\odot : \mathbb{F}_q^k \times \mathbb{F}_q^k \rightarrow \mathbb{F}_q^k$ :

- “Parallel multiplication”:

$$(a_1, a_2, \dots, a_k) \odot (b_1, b_2, \dots, b_k) = (a_1 b_1, a_2 b_2, \dots, a_k b_k).$$

- “Extension field multiplication”:

$$(a_0, a_1, \dots, a_{k-1}) \odot (b_0, b_1, \dots, b_{k-1}) = (c_0, c_1, \dots, c_{k-1}),$$

where these vectors are subject to the following relation. Let  $x, y, z \in \mathbb{F}_q[\alpha]$  with  $[\mathbb{F}_q[\alpha] : \mathbb{F}_q] = k$  be such that

$$x = \sum_{i=0}^{k-1} a_i \alpha^i,$$

$$y = \sum_{i=0}^{k-1} b_i \alpha^i$$

and

$$z = \sum_{i=0}^{k-1} c_i \alpha^i.$$

Then  $xy = z \in \mathbb{F}_q[\alpha]$ .

### 8.1.1 Parallel Multiplication

The first strongly multiplicative ramp scheme we discuss is due to Franklin and Yung [34]. When used for secure multi-party computation it has the advantage that, at the price of an additive factor  $k$  in the corruption tolerance, we can perform multiplication for  $k$  elements in parallel at the cost of a single multiplication. This leads to a multiplicative amortized cost reduction when a function need to be computed on many different inputs.

The ramp scheme is defined as follows. Let  $t$  and  $k$  be such that  $t + k - 1 < n$  and assume that the finite field  $\mathbb{F}_q$  is such that  $|\mathbb{F}_q| \geq n + k$ . Let the sets  $\{x_1, \dots, x_n\}$  and  $\{e_1, \dots, e_k\}$  be two disjoint sets of distinct elements from  $\mathbb{F}_q$ . Now to perform secret sharing with this scheme for a vector  $a = (u_1, \dots, u_k)$  of secret elements from  $\mathbb{F}_q$ , we select a random polynomial  $f(X) \in \mathbb{F}_q[X]$  of degree at most  $t + k - 1$  such that  $f(e_j) = u_j$  for  $j = 1, 2, \dots, k$  and define the shares to be  $a_j = f(x_j)$  for  $j = 1, 2, \dots, n$ .

Clearly,  $t + k$  shares or more jointly determine  $f$  and hence the secret vector  $a$ , so the access structure includes all player sets of size at least  $t + k$ . As to privacy, it is a straightforward consequence of Lagrange-interpolation that  $t$  or fewer shares jointly give no information on the secret vector, so the adversary structure includes all player sets of size at most  $t$ . We can sum these properties up by calling the resulting scheme a  $(t, t + k)$ -ramp scheme, with secrets of length  $k$ .

Assume that we additionally have a secret vector  $b = (v_1, \dots, v_k)$  with shares  $b_1, b_2, \dots, b_n$ , corresponding with a unique polynomial  $g(X) \in \mathbb{F}_q[X]$  of degree at most  $t + k - 1$ . Since for  $j = 1, 2, \dots, k$  it holds that  $(fg)(e_j) = u_j v_j$  and furthermore  $(fg)(x_i) = f(x_i)g(x_i)$  for  $i = 1, 2, \dots, n$ , it follows from Lagrange's interpolation theorem that the scheme is multiplicative for  $n > 2t + 2k - 1$  and  $\mathcal{A}$ -strongly multiplicative for  $n > 3t + 2k - 1$ , where  $\mathcal{A}$  consists of all subsets of  $\mathcal{N}$  of cardinality at most  $t$ .

### 8.1.2 Extension Field Multiplication

We now describe two ramp schemes that, when used for secure multi-party computation, allow to perform multiplication in an extension field of  $\mathbb{F}_q$  at the cost of a multiplication in  $\mathbb{F}_q$  when compared to Shamir's secret sharing scheme. Although the first scheme we describe here already achieves this effect, the second scheme has better parameters that are trivially seen to be optimal.

#### A First Attempt

The ramp scheme we describe first is due to Cramer, Damgård and de Haan [21]. With this ramp scheme it is possible to perform multiplications in a finite field using only communication and operations over a subfield, reducing the communication cost of every single multiplication by a multiplicative factor. For the technique to be used it is required that the finite field has a sufficiently large extension degree  $k$  over a subfield. Furthermore, the corruption tolerance needs to be decreased by an additive factor  $2k$ .

The scheme is defined as follows. Let  $t$  and  $k$  be such that  $t + 2k - 2 < n$ . A finite field  $\mathbb{F}_{q^k} = \mathbb{F}_q(\alpha)$  is selected such that  $|\mathbb{F}_q| > n$ . Let  $x_1, \dots, x_n$  be distinct non-zero elements from  $\mathbb{F}_q$ , let  $a = u_0 + u_1\alpha + \dots + u_{k-1}\alpha^{k-1} \in \mathbb{F}_{q^k}$  be a secret element and define  $u(X) = u_0 + u_1X + \dots + u_{k-1}X^{k-1} \in \mathbb{F}_q[X]$ . Choose a random polynomial  $r(X) \in \mathbb{F}_q[X]$  of degree at most  $t - 1$  and define  $f(X) = u(X) + r(X) \cdot X^{2k-1} \in \mathbb{F}_q[X]$ .

Clearly, since  $f$  has degree  $t + 2k - 2$ , it is clear that  $t + 2k - 1$  shares or more jointly determine  $f$  and hence the secret  $a$ . Therefore, the access structure contains all sets of cardinality  $t + 2k - 1$ . As for the adversary structure, let  $u'(X) \in \mathbb{F}_q[X]$  of degree at most  $k - 1$  be arbitrary and let  $r'(X)$  be the polynomial that evaluates to  $r(x_i) + (u(x_i) - u'(x_i))/x_i^{2k-1}$  for  $t$  points  $x_i$ . Then the polynomial  $f'(X) = u'(X) + r'(X) \cdot X^{2k-1}$  is consistent with the evaluation of  $f$  in these  $t$  points, but the secret corresponds with  $u'(X)$  here. So it is a  $(t, t + 2k - 1)$ -ramp scheme, with secrets of length  $k$ .

Now, when we multiply two such polynomials  $f(X) = u(X) + r(X) \cdot X^{2k-1}$  and  $g(X) = v(X) + r'(X) \cdot X^{2k-1}$ , the product polynomial  $fg$  has as its first  $2k - 1$  coefficients homogeneous sums  $H_\ell(a, b) = \sum_{i+j=\ell} u_i v_j$  of coefficients in  $u(X)$  and  $v(X)$  for  $\ell = 0, 1, \dots, 2k - 2$ . Furthermore, when  $n > 2t + 4k - 4$ , it is therefore an easy

consequence of Lagrange's interpolation theorem that for every such value  $H_\ell(a, b)$  there exist constants  $\mu_1^{(\ell)}, \mu_2^{(\ell)}, \dots, \mu_n^{(\ell)}$  such that  $H_\ell(a, b) = \sum_{i=0}^{2k} \mu_i^{(\ell)} (fg)(x_i)$ .

Since reducing an element  $\sum_{i=0}^{2k-2} c_i \cdot \alpha^i \in \mathbb{F}_q^k$  to its standard notation over the basis  $\{1, \alpha, \dots, \alpha^{k-1}\}$  can be done via linear operations on the coefficients  $c_i$ , we can now deduce the following lemma.

LEMMA 11. *There exist linear maps  $\chi_j : \mathbb{F}_q^{2k+1} \rightarrow \mathbb{F}_q$  ( $j = 0 \dots k$ ) such that for all  $a, b \in \mathbb{F}_{q^k}$*

$$ab = \sum_{j=0}^k \chi_j(H_0(a, b), \dots, H_{2k}(a, b)) \cdot \alpha^j,$$

where  $a$  and  $b$  are represented by their respective coordinate vectors  $(u_0, \dots, u_k)$  and  $(v_0, \dots, v_k)$ .

To summarize these results, if  $n > 2t + 4k - 4$  we can first determine the homogeneous sums  $H_0(a, b), H_1(a, b), \dots, H_k(a, b)$  via linear functions on the local share products  $(fg)(x_1), (fg)(x_2), \dots, (fg)(x_n)$ . By Lemma 11, we can then compute the coefficients of  $ab$  via linear functions on the values  $\{H_j(a, b)\}_{j=0}^k$ . Since applying multiple linear operations in turn preserves the linearity of the operations, it now follows that every coefficient of  $ab$  can be computed via a linear function on the local share products  $(fg)(x_1), (fg)(x_2), \dots, (fg)(x_n)$ . Therefore, the scheme is multiplicative for  $n > 2t + 4k - 4$  and in particular  $\mathcal{A}$ -strongly multiplicative for  $n > 3t + 4k - 4$ , where  $\mathcal{A}$  consists of all subsets of  $\mathcal{N}$  of cardinality at most  $t$ .

Note that in order to share a secret of length  $k$ , the scheme introduces a gap between the privacy and reconstruction thresholds of size  $2k - 1$ , whereas the scheme due to Franklin and Yung only requires a gap of size  $k$ . Below we introduce an improved version of this scheme that matches the latter thresholds.

### An Improved Version

A closer examination of the scheme above shows that it uses a secret sharing polynomial that has a fixed  $k$ -size gap between the lower degree coefficients that relate to the secret and the higher degree coefficients that introduce randomness. In fact, this explains the disparity between the parameters of the schemes described in Sections 8.1.1 and above.

The observation described in this section allows to remove this disparity and leads to a scheme with tight parameters that is additionally much easier to describe than the scheme from Section 8.1.2, while it achieves the same effect. Due to its more natural structure, it additionally generalizes over algebraic-geometric curves as demonstrated in Section 8.2.

The proposed scheme is based on the following theorem, which generalizes Lagrange's interpolation theorem to a setting where the evaluation points are taken from

## 8.1. Classical Ramp Schemes

---

different extension fields of a perfect base field  $K$  while the secret sharing polynomial is taken from  $K[X]$ . The idea is that the evaluation points get assigned different weights, depending on the extension degree of the smallest extension field of  $K$  in which they occur.

**THEOREM 34.** *Let  $K$  be a perfect field, and let  $\overline{K}$  denote an algebraic closure of  $K$ . Fix distinct  $a_1, \dots, a_\ell \in \overline{K}$  such that there is no pair  $a_i, a_j$  ( $i \neq j$ ) where  $a_j$  is a Galois-conjugate (over  $K$ ) of  $a_i$ . For  $i = 1, \dots, \ell$ , let  $n_i$  denote  $[K(a_i) : K]$ , the degree of  $K(a_i)$  over  $K$  as a field extension, and let  $N$  denote  $\sum_{i=1}^{\ell} [K(a_i) : K]$ . Then, for each  $b_1, \dots, b_\ell$  with  $b_i \in K(a_i)$  ( $i = 1, \dots, \ell$ ), there exists a unique polynomial  $f(X) \in K[X]$  such that  $\deg(f) < N$  and  $f(a_i) = b_i$ ,  $i = 1, \dots, \ell$ .*

**PROOF.** Let  $K[X]_{<N}$  denote the polynomials in  $K[X]$  of degree smaller than  $N$ . Consider the map

$$\phi : K[X]_{<N} \longrightarrow \bigoplus_{i=1}^{\ell} K(a_i), f \mapsto (f(a_1), \dots, f(a_\ell)).$$

We want to show that  $\phi$  is an isomorphism of  $K$ -vector spaces. Since the dimensions on both sides are equal, it is sufficient to argue that  $\phi$  is injective. Indeed, suppose  $g$  maps to 0. Then, for  $i = 1, \dots, \ell$ ,  $g(a_i) = 0$ . Since  $g \in K[X]$ ,  $g$  must be a multiple of the minimal polynomial  $h$  of  $a_i$  in  $K[X]$ . The Galois-conjugates of  $a_i$  are the roots of  $h$  and hence they are roots of  $g$ . Because the field is perfect,  $h$  is separable, i.e. all the roots of  $h$  are different, and the number of these roots is equal to  $n_i$ , so the number of conjugates of  $a_i$  is  $n_i$ . Note that  $a_i$  and  $a_j$  are not Galois conjugates for any  $i, j$  so  $g$  has at least  $\sum_{i=1}^{\ell} n_i = N$  zeroes in  $\overline{K}$ . Thus, viewing  $g$  as an element of  $\overline{K}[X]$ , we conclude that  $g \equiv 0$ .  $\triangle$

The new scheme is defined as follows. Let  $t$  and  $k$  be such that  $n > t + k - 1$ . A finite field  $\mathbb{F}_{q^k} = \mathbb{F}_q[\alpha]$  is selected such that  $|\mathbb{F}_q| \geq n$ . Let  $x_1, \dots, x_n$  be distinct (not necessarily non-zero) elements from  $\mathbb{F}_q$  and select  $e \in \mathbb{F}_{q^k}$  such that  $[\mathbb{F}_q(e) : \mathbb{F}_q] = k$ . The secret sharing is now performed as follows. For a secret element  $a \in \mathbb{F}_{q^k}$ , we choose a random polynomial  $f(X) \in \mathbb{F}_q[X]$  of degree at most  $t + k - 1$  such that  $f(e) = a$ . The shares are again  $f(x_1), f(x_2), \dots, f(x_n)$ .

**THEOREM 35.** *This defines a  $(t, t + k)$ -ramp scheme with secrets of length  $k$ .*

**PROOF.** Accepted sets: Given the value of  $f$  in  $t + k$  points  $x_{i_1}, \dots, x_{i_{t+k}}$ , we can apply the previous theorem with  $\ell = t + k$ ,  $a_j = x_{i_j}$  (so  $n_j = 1$  and  $N = t + k$ ), to see that these shares determine the polynomial and hence the secret.

Rejected sets: Given the value of  $f$  in  $t$  points  $x_{i_1}, \dots, x_{i_t}$  take in the previous theorem  $\ell = t + 1$ ,  $a_j = x_{i_j}$  for  $j = 1, \dots, \ell - 1$  and  $a_\ell = e$ . Then  $n_j = 1$  for  $j = 1, \dots, \ell - 1$  and  $n_\ell = k$ , so  $N = t + k$ . The theorem shows that for every possible

choice of the secret  $a \in \mathbb{F}_{q^k}$ , there exists a unique polynomial of degree less than  $t + k$  such that  $f(e) = a$  and  $f$  evaluates to the known values in  $x_{i_1}, \dots, x_{i_t}$ .  $\triangle$

One can additionally show that this scheme is multiplicative for  $n > 2t + 2k - 2$  and  $\mathcal{A}$ -strongly multiplicative for  $n > 3t + 2k - 2$ , where  $\mathcal{A}$  consists of all subsets of  $\mathcal{N}$  of cardinality at most  $t$ . Since this is proven for a generalized version of this scheme in Section 8.2.4, we omit the details here.

## 8.2 Algebraic-Geometric Ramp Schemes

We now demonstrate generalizations of the schemes discussed in the previous section based on algebraic-geometric coding techniques. This approach to constructing ramp schemes was initially invented by Chen and Cramer [15], who constructed the first family of strongly multiplicative ramp schemes that can be defined over fields of a size that is much smaller than the number of participants  $n$ , or even of constant-size. This family of ramp schemes can be seen as a direct generalization of Shamir's secret sharing scheme and allows for a straightforward extension that generalizes the ramp scheme of Franklin and Yung [34].

We first describe this generalization of the Franklin-Yung scheme, that corresponds with the parallel multiplication map, and then introduce its new counterpart for extension field multiplication that has recently appeared in work by Chen, Cramer, de Haan and Cascudo [16]. All algebraic-geometric ramp schemes that are presented here not only resolve the dependence of the field size on the number of participants  $n$ , but additionally resolve the dependence of the share sizes on the size of the secret, i.e., have a high information rate.

### 8.2.1 Interpolation in Riemann-Roch spaces

We require the following result, which can be seen as an extension of Lagrange's interpolation theorem for Riemann-Roch spaces. It is the algebraic-geometric counterpart of Theorem 34 defined for arbitrary smooth projective curves  $\mathcal{C}$  defined over  $\mathbb{F}_q$ .

**THEOREM 36.** *Let  $P_1, \dots, P_\ell$  be points on a smooth projective curve  $\mathcal{C}$  defined over  $\mathbb{F}_q$  such that  $P_i$  and  $P_j$  are not Galois-conjugate for any  $i \neq j$  and let  $G$  be an  $\mathbb{F}_q$ -rational divisor such that  $\text{supp}(G) \cap \{P_1, \dots, P_\ell\} = \emptyset$ . Furthermore, for  $i \in \{1, \dots, \ell\}$  let  $n_i$  be the smallest number such that  $P_i$  is  $\mathbb{F}_{q^{n_i}}$ -rational and define  $N = \sum_{i=1}^{\ell} n_i$ . Then:*

1. *If  $N \geq \text{deg}(G) + 1$ , then for any vector  $(y_1, \dots, y_\ell)$  with  $y_i \in \mathbb{F}_{q^{n_i}}$  there exists at most one  $f \in L(G)$  such that  $f(P_i) = y_i$  for all  $i \in \{1, 2, \dots, \ell\}$ .*
2. *If  $N \leq \text{deg}(G) - 2g + 1$ , then for any vector  $(y_1, \dots, y_\ell)$  with  $y_i \in \mathbb{F}_{q^{n_i}}$  there exists at least one  $f \in L(G)$  such that  $f(P_i) = y_i$  for all  $i \in \{1, 2, \dots, \ell\}$ . Furthermore,*

## 8.2. Algebraic-Geometric Ramp Schemes

---

the number of such  $\mathbb{F}_q$ -rational functions is the same for any vector  $(y_1, \dots, y_\ell)$  with  $y_i \in \mathbb{F}_{q^{n_i}}$ .

PROOF. Let the map

$$\phi : L(G) \rightarrow \bigoplus_{i=1}^{\ell} \mathbb{F}_{q^{n_i}}$$

be defined by

$$f \mapsto (f(P_1), \dots, f(P_\ell)).$$

Furthermore, for  $i \in \{1, 2, \dots, \ell\}$  let  $P_i^{(0)} = P_i, \dots, P_i^{(n_i-1)}$  be the  $n_i$  Galois-conjugates of  $P_i$  under the Frobenius automorphism over  $\mathbb{F}_q$ . Observe that the point  $\sum_{j=0}^{n_i-1} P_i^{(j)}$  is  $\mathbb{F}_q$ -rational, as any element of the group  $\text{Gal}(\overline{\mathbb{F}}_q/\mathbb{F}_q)$  permutes the conjugates of  $P_i$ .

Now define the divisor

$$A = G - \sum_{i=1}^{\ell} \left( \sum_{j=0}^{n_i-1} P_i^{(j)} \right).$$

Then  $\text{Ker}(\phi) = L(A)$ . Observe that  $\text{deg}(A) = \text{deg}(G) - N$ . Then

1. If  $N \geq \text{deg}(G) + 1$ ,  $\text{deg}(A) < 0$  and  $\ell(A) = 0$ . Hence  $\phi$  is injective, which proves the property.
2. If  $N \leq \text{deg}(G) - 2g + 1$ , then  $\text{deg}(A) \geq 2g - 1$  and we can invoke the Riemann-Roch Theorem to conclude that

$$l(A) = \text{deg}(A) - g + 1 = \text{deg}(G) - N - g + 1 = l(G) - N.$$

Furthermore,

$$\dim(\text{Im}(\phi)) = \dim(L(G)) - \dim(\text{Ker}\phi) = l(G) - l(A) = N.$$

Therefore  $\phi$  is surjective.

△

### 8.2.2 Parallel Multiplication

Let  $D = \{Q_1, \dots, Q_k, P_1, \dots, P_n\}$  be a set of  $\mathbb{F}_q$ -rational points on a smooth projective curve  $\mathcal{C}$  defined over  $\mathbb{F}_q$  and let  $G$  be an  $\mathbb{F}_q$ -rational divisor of degree  $2g + t + k - 1$  with support disjoint from  $D$ . Note that since  $G$  can have support outside the  $\mathbb{F}_q$ -rational points, it is possible to include all  $\mathbb{F}_q$ -rational points on  $\mathcal{C}$  in  $D$ .

Every point  $P_i$  corresponds with an  $i^{\text{th}}$  share and every point  $Q_j$  corresponds to the  $j^{\text{th}}$  position of a secret vector, as follows. For any secret vector  $(s_1, \dots, s_k) \in \mathbb{F}_q^k$  a rational function  $f \in L(G)$  is selected uniformly at random under the restriction that  $f(Q_j) = s_j$  for all  $j \in \{1, \dots, k\}$ . For any  $i \in \mathcal{N}$ , the  $i^{\text{th}}$  share corresponding to the secret is defined to be the value  $f(P_i) \in \mathbb{F}_q$ .

We note that the scheme described above fits into the formal matricial definition of ramp scheme given in Definition 32. Let  $\{f_1, \dots, f_u\}$  be a basis of  $L(G)$  such that  $f_i(Q_j) = 1$  if  $i = j$  and  $f_i(Q_j) = 0$  if  $i \neq j$ , for  $i \in \{1, \dots, u\}$  and  $j \in \{1, \dots, k\}$ . It is easy to see that we can always choose such a basis due to Theorem 36. Indeed, we have that  $k < \deg(G) - 2g + 1 = t + k + 1$  so the Theorem 36 ensures the existence of such  $f_i$  for  $i = 1, \dots, k$ . Now simply take  $\{f_{k+1}, \dots, f_u\}$  as a basis of  $L(G - \sum_{i=1}^k Q_i)$ , which has dimension  $u - k$  according to the Riemann-Roch Theorem.

Now, define  $M$  to be the matrix

$$M = \begin{pmatrix} f_1(Q_1) & f_2(Q_1) & \dots & f_u(Q_1) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(Q_k) & f_2(Q_k) & \dots & f_u(Q_k) \\ f_1(P_1) & f_2(P_1) & \dots & f_u(P_1) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(P_n) & f_2(P_n) & \dots & f_u(P_n) \end{pmatrix}.$$

Then it is now easy to verify that the first  $k$  rows correspond with the first  $k$  unit vectors in  $\mathbb{F}_q^u$ . Furthermore, if we take a random  $\mathbb{F}_q$ -rational function  $g \in L(G)$  where

$$g = \sum_{j=1}^k s_j f_j + \sum_{j=k+1}^n r_j f_j,$$

evaluating  $g$  in the points  $Q_1, \dots, Q_k, P_1, \dots, P_n$  corresponds with multiplication of  $M$  with the vector  $\vec{v} = (s_1, \dots, s_k, r_{k+1}, \dots, r_n)$ . In particular, multiplying any row of  $M_i$  by  $\vec{v}$  corresponds with calculating  $g(P_i)$ . Similarly, it holds that  $g(Q_j) = s_j$  for any  $j \in \{1, \dots, k\}$ .

**THEOREM 37.** *This defines a  $(t, 2g + t + k)$ -ramp scheme  $(\mathbb{F}_q, M, Id_{\mathcal{N}})$  with secrets of length  $k$ .*

**PROOF.** It can be easily seen as a special case of Theorem 36 that any rational function in  $L(G)$  is uniquely determined by its evaluations in  $\deg(G) + 1$  rational points (this is exactly Lemma 1 of [15]). In this case,  $\deg(G) = 2g + t + k - 1$  so any  $2g + t + k$  shares determine the rational function and thus the secret vector.

Now let  $A$  be any subset of  $\mathcal{N}$  of cardinality  $t$ . We only need to argue that for any secret vector  $\vec{s} = (s_1, \dots, s_k) \in \mathbb{F}_q^k$  there exists a rational function  $f$  such that

$f(Q_i) = s_i$  and the evaluation of  $f$  in the points  $\{P_i\}_{i \in A}$  is zero. This follows from Theorem 36, since  $t + k = \deg(G) - 2g + 1$ .  $\triangle$

### 8.2.3 Extension Field Multiplication

Let  $D = \{P_1, \dots, P_n\}$  be a set of  $\mathbb{F}_q$ -rational points on a smooth projective curve  $\mathcal{C}$  defined over  $\mathbb{F}_q$  and let  $G$  be an  $\mathbb{F}_q$ -rational divisor of degree  $2g + t + k - 1$  with support disjoint from  $D$ . Furthermore, assume that  $Q$  is a point on the curve outside the support of  $G$  that is  $\mathbb{F}_{q^k}$ -rational and not  $\mathbb{F}_{q^d}$ -rational for any integer  $d < k$ .

Let  $\{e_1, e_2, \dots, e_k\}$  be a basis of  $\mathbb{F}_{q^k}$  over  $\mathbb{F}_q$ . To share a secret vector  $(s_1, \dots, s_k)$ , a rational function  $f \in L(G)$  is selected uniformly at random under the restriction that  $f(Q) = s_1 e_1 + \dots + s_k e_k \in \mathbb{F}_{q^k}$ . The shares are defined to be the values  $f(P_i) \in \mathbb{F}_q$  for  $i \in \mathcal{N}$ .

We can also represent this ramp scheme in the standard matricial form. In this case we take a basis  $\{f_1, \dots, f_u\}$  of  $L(G)$  such that  $f_i(Q) = e_i$  for  $i = 1, \dots, k$  and  $f_i(Q) = 0$  for  $i = k + 1, \dots, u$ . It can again be shown that such a basis exists using Theorem 36. We have only one point of degree  $k$  and  $k \leq \deg(G) - 2g + 1$ , so we know such  $f_i$  exist for  $i = 1, \dots, k$ , and we can take  $\{f_{k+1}, \dots, f_u\}$  a basis of  $L(G - Q - \sum_{i=1}^{k-1} Q_i)$ , where  $Q_1, Q_2, \dots, Q_{k-1}$  are the conjugate points of  $Q$  under the Frobenius automorphism over  $\mathbb{F}_q$ .

Let  $M$  be the matrix

$$M = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & \dots & 0 \\ f_1(P_1) & f_2(P_1) & f_3(P_1) & \dots & f_i(P_1) & \dots & f_u(P_1) \\ f_1(P_2) & f_2(P_2) & f_3(P_2) & \dots & f_i(P_2) & \dots & f_u(P_2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_1(P_n) & f_2(P_n) & f_3(P_n) & \dots & f_i(P_n) & \dots & f_u(P_n) \end{pmatrix}.$$

As before, if we take a random  $\mathbb{F}_q$ -rational function  $g \in L(G)$  where

$$g = \sum_{j=1}^k s_j f_j + \sum_{j=k+1}^n r_j f_j,$$

evaluating  $g$  in the points  $Q, P_1, \dots, P_n$  corresponds with multiplication of  $M$  with the vector  $\vec{v} = (s_1, \dots, s_k, r_{k+1}, \dots, r_n)$ . In particular, multiplying any row of  $M_i$  by  $\vec{v}$  corresponds again with calculating  $g(P_i)$ . Similarly, it holds under this construction that  $g(Q) = \sum_{i=1}^k s_i e_i$  by the choice of the basis.

**THEOREM 38.** *This defines a  $(t, 2g + t + k)$ -ramp scheme  $(\mathbb{F}_q, M, Id_{\mathcal{N}})$  with secrets of length  $k$ .*

**PROOF.** As before, both properties are a direct consequence of Theorem 36.  $\triangle$

### 8.2.4 Multiplication Properties

The schemes thus described correspond with the two listed multiplication maps. For the parallel multiplication scheme, given two vectors  $\vec{s} = (s_1, s_2, \dots, s_k)$  and  $\vec{t} = (t_1, t_2, \dots, t_k)$ , the product is  $\vec{s} \odot \vec{t} = (s_1 t_1, s_2 t_2, \dots, s_k t_k)$ .

For the extension field multiplication scheme, for any vectors  $\vec{s} = (s_1, s_2, \dots, s_k)$  and  $\vec{t} = (t_1, t_2, \dots, t_k)$ , representing the elements  $s = s_1 e_1 + s_2 e_2 + \dots + s_k e_k \in \mathbb{F}_{q^k}$  and  $t = t_1 e_1 + t_2 e_2 + \dots + t_k e_k \in \mathbb{F}_{q^k}$ , the product of these two elements in the field  $\mathbb{F}_{q^k}$  is some element  $u = u_1 e_1 + u_2 e_2 + \dots + u_k e_k \in \mathbb{F}_{q^k}$  for some  $u_i \in \mathbb{F}_q$ . We can therefore define the product of  $\vec{s}$  and  $\vec{t}$  as  $\vec{s} \odot \vec{t} = (u_1, u_2, \dots, u_k)$ .

We next prove that, if  $n$  is large enough with respect to  $t$ , the two schemes are (strongly) multiplicative with regard to their respective multiplications.

**THEOREM 39.** *The parallel multiplication scheme is multiplicative when  $n \geq 2t + 4g + 2k - 1$  and  $\mathcal{A}$ -strongly multiplicative with respect to a  $t$ -adversary  $\mathcal{A}$  when  $n \geq 3t + 4g + 2k - 1$ .*

**PROOF.** We need to show that for any  $i = 1, \dots, k$  there exist coefficients  $\lambda_1^{(i)}, \dots, \lambda_n^{(i)}$  such that for any  $f, g \in L(G)$ ,

$$f(Q_i)g(Q_i) = \sum_{j=1}^n \lambda_j^{(i)} f(P_j)g(P_j).$$

Note that if  $f$  and  $g$  are in  $L(G)$  their product is in the space  $L(2G)$ .

According to Theorem 36 we have that if  $\deg(2G) + 1 \leq n$  the map

$$\phi : L(2G) \rightarrow \bigoplus_{j=1}^n \mathbb{F}_q$$

defined by

$$h \mapsto (h(P_1), \dots, h(P_n))$$

is linear and injective, so it has an inverse and it is also linear. Furthermore, the maps

$$\psi_i : L(2G) \rightarrow \mathbb{F}_q$$

defined by

$$h \mapsto h(Q_i)$$

## 8.2. Algebraic-Geometric Ramp Schemes

---

are also linear for any  $i \in \{1, \dots, k\}$ . So the composition of  $\phi^{-1}$  and any  $\psi_i$  is linear. Therefore  $(fg)(Q_i)$  is a linear combination of  $f(P_j)g(P_j)$  for any  $f$  and  $g$  in  $L(G)$ . Finally observe that the condition  $\deg(2G)+1 \leq n$  holds whenever  $4g+2t+2k-1 \leq n$ .  $\triangle$

Similar to the simpler finite field setting the coefficients  $\lambda_j^{(i)}$  can be explicitly determined. We now describe how to obtain these using the Residue Theorem [75].

### Determining the coefficients $\lambda_j^{(i)}$ .

A consequence of the Residue Theorem is that for any function  $\varphi$  in  $L(2G)$  and any differential  $\omega$  in  $\Omega(Q_i + \sum_{j=1}^n P_j - 2G)$  the relation

$$\begin{aligned} 0 &= \sum_{j=1}^n \text{res}_{P_j}(\varphi\omega) + \text{res}_{Q_i}(\varphi\omega) \\ &= \sum_{j=1}^n \varphi(P_j)\text{res}_{P_j}(\omega) + \varphi(Q_i)\text{res}_{Q_i}(\omega) \end{aligned}$$

holds. Therefore, if there exists a nonzero element  $\omega$  in  $\Omega(Q_i + \sum_{j=1}^n P_j - 2G)$ , applying the theorem for the rational function  $fg$  gives a linear relation between the values  $fg(Q_i)$  and  $fg(P_j)$  for  $j = 1, \dots, n$  for some coefficients which do not depend on  $f$  and  $g$ . If we can additionally ensure that the coefficient  $\text{res}_{Q_i}(\omega)$  is non-zero, then we have a relation of the form

$$fg(Q_i) = \sum_{j=1}^n -\frac{\text{res}_{P_j}(\omega)}{\text{res}_{Q_i}(\omega)} fg(P_j).$$

Thus,

$$\lambda_j^{(i)} = -\frac{\text{res}_{P_j}(\omega)}{\text{res}_{Q_i}(\omega)}$$

and we are done.

It is a known fact that we can define an isomorphism of  $\mathbb{F}_q$ -vector spaces

$$\phi : L(K + Q_i + \sum_{j=1}^n P_j - 2G) \rightarrow \Omega(Q_i + \sum_{j=1}^n P_j - 2G)$$

defined by

$$\phi(h) = h\eta$$

where  $K$  is a canonical divisor and  $\eta$  is a differential such that  $\text{div}(\eta) = K$ . It suffices to find an element  $h$  in  $L(K + Q_i + \sum_{j=1}^n P_j - 2G)$  with a first order pole in  $Q_i$ . Hence, we have to show that there exists an element in the difference of the spaces

$L(K + Q_i + \sum_{j=1}^n P_j - 2G)$  and  $L(K + \sum_{j=1}^n P_j - 2G)$ . Applying the Riemann-Roch theorem for  $n \geq 2t + 4g + 2k - 1$  shows us that the dimensions of these spaces differ and the result follows.

**THEOREM 40.** *The extension field multiplication scheme is multiplicative when  $n \geq 2t + 4g + 2k - 1$  and  $\mathcal{A}$ -strongly multiplicative with respect to a  $t$ -adversary  $\mathcal{A}$  when  $n \geq 3t + 4g + 2k - 1$ .*

**PROOF.** Now we need to show that for any  $i = 1, \dots, k$  there exist coefficients  $\lambda_1^{(i)}, \dots, \lambda_n^{(i)}$  in  $\mathbb{F}_q$  such that for any  $f, g \in L(G)$ ,

$$\pi_i(f(Q)g(Q)) = \sum_{j=1}^n \lambda_j^{(i)} f(P_j)g(P_j).$$

An argument similar to that in Theorem 39 shows that, for  $n \geq 2t + 4g + 2k - 1$ , there exist elements  $r_j \in \mathbb{F}_{q^k}$  such that  $f(Q)g(Q) = \sum_{j=1}^n r_j f(P_j)g(P_j)$ . Now, note that  $r_j = \sum_{i=1}^k \lambda_j^{(i)} e_i$ , which gives us the desired result.  $\triangle$

### 8.2.5 Achieving Constant Field Size

For both of the algebraic-geometric ramp schemes defined above, the actual parameters of the scheme depend on the parameters of the curve that is used. In particular, for the case where  $g = 0$  the schemes collapse into the classical schemes described in Section 8.1.

Recall that strongly multiplicative secret sharing schemes only exist for adversary structures that are  $Q^{(3)}$ , which for strongly multiplicative  $t$ -threshold schemes implies that  $t < n/3$ . We now demonstrate that when one bases the algebraic-geometric ramp schemes on appropriate curves, it is possible to construct families of strongly multiplicative ramp schemes over constant size fields rejecting all player sets of size  $t$  that additionally have close to optimal parameters, i.e., such that for any  $\epsilon > 0$  one can achieve  $(1/3 - \epsilon)n < t < n/3$ .

As already noted in [15], one currently obtains the best parameters for the scheme with parallel multiplication when using the curves of Garcia and Stichtenoth [36]. For a square  $q$ , they define a family of curves  $\{C_i\}_{i \in \mathbb{Z}_{\geq 0}}$  defined over  $\mathbb{F}_q$  with

$$\lim_{i \rightarrow \infty} \frac{\#C_i(\mathbb{F}_q)}{g(C_i)} = \sqrt{q} - 1,$$

i.e., that attains the Drinfeld-Vlăduț bound.

Suppose we take a curve  $C_m$  in the family that is very close to optimal, i.e., such that  $\#C_m(\mathbb{F}_q)/g(C_m) = (1 - \gamma)(\sqrt{q} - 1)$  for some constant  $\gamma > 0$  that can be chosen arbitrarily small. The parameter  $n$  can be as large as the number of  $\mathbb{F}_q$ -rational points

## 8.2. Algebraic-Geometric Ramp Schemes

---

on the curve minus the number  $k$  of evaluation points used for the coefficients of the secret vector, which implies that we can take  $t$  maximal such that

$$t < \left(\frac{1}{3} - c\right) \cdot n,$$

where

$$c = \frac{4}{3(1-\gamma)(\sqrt{q}-1)} - \frac{4k}{3n(1-\gamma)(\sqrt{q}-1)} + \frac{2k-2}{3n} < \frac{1}{3}.$$

So for any  $0 < \epsilon < 1/6$  there exists a finite field  $\mathbb{F}_q$  with  $q \geq 49$  such that for infinitely many  $n$  there exists a strongly multiplicative scheme defined over  $\mathbb{F}_q$  tolerating a  $t$ -adversary with  $(1/3 - \epsilon)n \leq t \leq n/3$ .<sup>1</sup>

For the schemes with extension field multiplication the same result is achieved by these curves, except that we additionally need to ensure that we can select at least one  $\mathbb{F}_{q^k}$ -rational point on every curve in the family. The main situation we are interested in, since this allows to achieve the best communication when considering multi-party protocols based on these algebraic-geometric schemes, is that where the extension degree  $k = \delta n$  for some constant value  $0 < \delta < 1$ . In particular, for the asymptotic result we are describing we can assume that  $\delta$  is arbitrarily small.

By a corollary of the Hasse-Weil bound (see for instance Theorem V.2.10 in [72]), a smooth, projective curve of genus  $g$  defined over  $\mathbb{F}_q$  must have an  $\mathbb{F}_{q^k}$ -rational point for any  $k$  that satisfies  $k \geq 4g + 3$ . If we pick a large enough (square)  $q$ , select a curve  $C_m$  as before and set  $n = \#C_m(\mathbb{F}_q)$ , we can ensure that

$$k = \delta n \geq \delta(1-\gamma)(\sqrt{q}-1)g(C_m) > 4g(C_m) + 3.$$

If we let  $n$  be equal to the total number of  $\mathbb{F}_q$ -rational points on the curve, this implies that we can take  $t$  maximal such that

$$t < \left(\frac{1}{3} - c\right) \cdot n,$$

where

$$c = \frac{4}{3(1-\gamma)(\sqrt{q}-1)} + \frac{2\delta}{3} - \frac{2}{3n} < \frac{1}{3}.$$

So for any  $0 < \epsilon < 1/6$  there exist a sufficiently small  $\delta > 0$  with  $k = \delta n$  and a finite field  $\mathbb{F}_q$  with  $q \geq 49$  such that for infinitely many  $n$  there exists a strongly multiplicative scheme defined over  $\mathbb{F}_q$  tolerating a  $t$ -adversary with  $(1/3 - \epsilon)n \leq t \leq n/3$ . This family is particularly interesting in that it introduces a “constant overhead”, i.e., the total sum of the share sizes is  $\Omega(n)$  while the secret is also of size  $\Omega(n)$ .

---

<sup>1</sup>Note that in order to theoretically achieve this result it actually already suffices to have a limit  $\lim_{i \rightarrow \infty} \#C_i(\mathbb{F}_q)/g(C_i)$  that increases unbounded with  $q$ , although this will in general require a much larger value of  $q$ . In fact, the 1985 result by Serre [65] that states that for all  $q$  it holds that  $A(q) > c \cdot \log q$  for a certain positive absolute constant  $c$  already suffices here.



Part IV

Protocols



## Chapter 9

# Basics of Secure Multi-Party Computation

It is well-known in the literature how to use (strongly) multiplicative secret sharing schemes to construct multi-party computation protocols secure against a passive (active) adversary. However, in order to achieve the same effect using (strongly) multiplicative ramp schemes the known techniques require various adaptations. We describe these adaptations in this and the next chapter.

We recall that, in a nutshell, secure multi-party computation concerns  $n$  players  $p_1, p_2, \dots, p_n$  that each hold their own respective private input vector  $\vec{y}_i$  consisting of a finite number of elements from a finite field  $\mathbb{F}_q$  for  $i = 1, 2, \dots, n$ . The goal of the computation is for the players to together determine the output of a given function applied to the inputs under the presence of a passive or active adversary that can corrupt some of the players, while keeping the inputs  $\vec{y}_1, \vec{y}_2, \dots, \vec{y}_n$  as private as possible and guaranteeing correctness of the output.

In this chapter we for completeness provide a brief glimpse of some fundamental techniques that are at the base of general perfectly secure multi-party computation. In essence, we have taken the most basic known techniques that are used to construct multi-party computation protocols secure against a passive adversary from multiplicative linear secret sharing schemes and modified these for use with multiplicative ramp schemes.

Although we do not discuss the security for these protocols in detail, we do point out how one would go about formally defining it. The protocols described in this and the following chapter can be shown to remain secure against the same type of adversary as is handled by the underlying ramp scheme.

## 9.1 Protocol Structure

We first sketch the general structure of multi-party computation protocols that are constructed from (strongly) multiplicative secret sharing schemes. These protocols consist of the following components.

**Initialization** Every player secret shares his input among the players using the underlying secret sharing scheme.

**Computation** Based on the arithmetic circuit of the given function, the function value is computed step-by-step via consecutive operations on the shares. That is, whenever a gate in the circuit requires an addition or multiplication involving one or two intermediate values the players execute a corresponding operation on the shares. This results in new shares in the secret sharing scheme for the players with as corresponding secret the output value of the gate.

**Output** All players broadcast their share in the output of the last gate in the circuit, which enables the players to determine the function value corresponding with their inputs.

From this description it is obvious that, given a multiplicative secret sharing scheme, the execution of the protocol mainly depends on the specifications of the addition and multiplication operations.

## 9.2 Addition and Multiplication in the Passive Case

We now describe how one can perform the addition and multiplication operations on secrets using the corresponding shares in a multiplicative ramp scheme. Let  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$  be a ramp scheme that is multiplicative with respect to the multiplication operation  $\odot$ . For simplicity we assume that every share in the scheme consists of a single value in  $\mathbb{F}_q$ , i.e.,  $M$  is an  $(n+k) \times e$ -matrix and  $\psi$  is the identity map, and remark that the general case proceeds similarly.

Suppose that  $\vec{s} = (s_i)_{i=1}^k \in \mathbb{F}_q^k$  is a secret vector with corresponding shares  $a_1, a_2, \dots, a_n$  and put  $\vec{x} = (s_1, \dots, s_k, a_1, \dots, a_n)$ . Then there exists a vector

$$\vec{v} = (s_1, \dots, s_k, r_{k+1}, \dots, r_e) \in \mathbb{F}_q^k$$

such that  $\vec{x} = M \cdot (\vec{v})^T$ . Vice versa, to demonstrate that certain values  $a'_1, \dots, a'_n$  are shares in a secret vector  $\vec{s}' = (s'_1, \dots, s'_k)$  it suffices to show that there exist values  $r'_{k+1}, \dots, r'_e \in \mathbb{F}_q$  such that

$$(s'_1, \dots, s'_k, a'_1, \dots, a'_n) = M \cdot (s'_1, \dots, s'_k, r'_{k+1}, \dots, r'_e)^T.$$

We will use this notation throughout this section. The next part describes the general operations on shares in a multiplicative ramp scheme that allow to perform the required computation steps. Since these operations are very similar to their well-known counterparts for linear secret sharing schemes, we omit the details concerning the privacy and correctness conditions surrounding these operations and refer the interested reader to [23].

### Addition of a constant vector

Let a constant vector  $\vec{c} = (c_1, c_2, \dots, c_k) \in \mathbb{F}_q^k$  be given and suppose the current gate in the circuit requires the players to compute shares in  $\vec{s} + \vec{c}$ , where the players hold the shares  $a_1, \dots, a_n$  in  $\vec{s}$ .

Suppose

$$\vec{\ell} = (c_1, \dots, c_k, \ell_1, \dots, \ell_n) = M \cdot (c_1, \dots, c_k, 0, \dots, 0)^T,$$

i.e.,  $\ell_1, \dots, \ell_n$  are given shares for the “secret”  $\vec{c}$ . Then the values  $(a_i + \ell_i)_{i=1}^n$  are shares in  $\vec{s} + \vec{c}$ , since for the vector

$$\vec{x} + \vec{\ell} = (s_1 + c_1, \dots, s_k + c_k, a_1 + \ell_1, \dots, a_n + \ell_n)$$

it holds that

$$\begin{aligned} \vec{x} + \vec{\ell} &= M \cdot (\vec{v})^T + M \cdot (c_1, c_2, \dots, c_k, 0, \dots, 0)^T \\ &= M \cdot (\vec{v} + (c_1, c_2, \dots, c_k, 0, \dots, 0))^T \\ &= M \cdot (s_1 + c_1, \dots, s_k + c_k, r_{k+1}, \dots, r_e)^T. \end{aligned}$$

In other words, every player  $p_i$  can compute the value  $a_i + \ell_i$  using his local share  $a_i$  in  $\vec{s}$  and the public value  $\ell_i$ . This results in distributed shares in the secret  $\vec{s} + \vec{c}$ . Note that these shares can be computed without interaction between the players.

### Addition of two secret vectors

Suppose that in addition to the shares  $a_1, \dots, a_n$  in a secret vector  $\vec{s}$  the players hold shares  $b_1, \dots, b_n$  in a secret vector  $\vec{t}$ , where the current gate in the circuit requires the players to compute shares in  $\vec{s} + \vec{t}$ . A similar argument as above shows that the values  $\{a_i + b_i\}_{i=1}^n$  are shares in  $\vec{s} + \vec{t}$ , which can again be computed locally by every player without interaction.

### Multiplication with a constant value

Let a constant value  $c \in \mathbb{F}_q$  be given and suppose the current gate in the circuit requires the players to compute shares in  $c \cdot \vec{s}$ , where the players hold the shares

$a_1, \dots, a_n$  in  $\vec{s}$ . Then the values  $\{ca_i\}_{i=1}^n$  are shares in  $c \cdot \vec{s}$ , since

$$\begin{aligned} (cs_1, \dots, cs_k, ca_1, \dots, ca_n) &= c(M \cdot (\vec{v})^T) \\ &= M \cdot (c \cdot \vec{v})^T \\ &= M \cdot (cs_1, \dots, cs_k, cr_{k+1}, \dots, cr_e)^T. \end{aligned}$$

These shares can again be computed without interaction by the players.

### Multiplication of two secret vectors

Suppose that the players hold the shares  $a_1, \dots, a_n$  in a secret vector  $\vec{s}$  and shares  $b_1, \dots, b_n$  in a secret vector  $\vec{t}$ , where the current gate in the circuit requires the players to compute shares in  $\vec{s} \odot \vec{t}$ . Secure multiplication of these two secret vectors is handled via a reduction of secure multiplication to the previously resolved problem of secure addition.

Since the ramp scheme  $\mathcal{M}$  is multiplicative, there exist vectors

$$\vec{\lambda}_j = (\lambda_j^{(1)}, \dots, \lambda_j^{(k)}) \in \mathbb{F}_q^k$$

for  $j = 1, \dots, n$  such that

$$\vec{s} \odot \vec{t} = \sum_{j=1}^n \vec{\lambda}_j a_j b_j.$$

Now suppose for  $i = 1, 2, \dots, n$  every player  $p_i$  secret shares the vector  $\vec{\lambda}_i a_i b_i$ , where the value  $a_i b_i$  can be computed locally by player  $p_i$  and the vector  $\vec{\lambda}_i$  is given. Let this result in shares  $d_{i1}, d_{i2}, \dots, d_{in}$  in the vector  $\vec{\lambda}_i a_i b_i$  for  $i = 1, 2, \dots, n$ . Then by the previously treated case of secure addition, it holds that the locally computable values  $\sum_{i=1}^n d_{i1}, \sum_{i=1}^n d_{i2}, \dots, \sum_{i=1}^n d_{in}$  are shares in the vector  $\sum_{j=1}^n \vec{\lambda}_j a_j b_j = \vec{s} \odot \vec{t}$ .

Due to the  $n$  applications of secret sharing required for secure multiplication, one for each player, the multiplication gates are the only computation steps where communication is required.

## 9.3 Low Communication MPC with Ramp Schemes

The amount of communication involved for the computation steps typically dominates the communication required for the initialization and output steps, which makes it crucial to keep the communication required for secure multiplications low. We here give an indication of the differences in communication required for one secure multiplication when one considers protocols secure against a passive  $t$ -adversary based on different ramp schemes, where  $t < (1/2 - \epsilon)n$  with  $\epsilon > 0$  small.

Previously, one of the standard ramp schemes used for most practical applications in multi-party computation is Shamir's secret sharing scheme. When one uses this scheme, the communication involved for a multiplication amounts to  $n^2$  field elements, where the field size additionally needs to be at least equal to the number of players involved. This means that when we measure the amount of communication required in bits, every secure multiplication requires  $\Omega(n^2 \log n)$  bits.

If one instead uses the coding theoretical constructions from Chapter 7 or the algebraic-geometric ramp scheme by Chen and Cramer, one can drop the dependence between the field size and the number of players, which reduces the amount of required communication for a secure multiplication to  $\Omega(n^2)$  bits for a constant-size secret. Using the algebraic-geometric ramp scheme with extension field multiplication from Chapter 8, this can theoretically be reduced to  $\Omega(n^2)$  bits communication for a secret consisting of  $\Omega(n)$  bits in an extension field  $\mathbb{F}_{q^k}$ , where  $q$  is constant and  $k$  is linear in  $n$ .

To the best of our knowledge, there are no other techniques known that make communication improvements at the level of a single (perfectly) secure multiplication. However, when one considers secure computations as a whole and looks at subclasses of functions that are more structured, one can do better.

For instance, when many instances of the same function need to be computed securely at the same time, one can group multiplications. Using the ramp scheme of Franklin and Yung, one can achieve communication of  $\Omega(n^2 \log n)$  bits for  $\Omega(n)$  multiplications. This is improved by the algebraic-geometric ramp scheme with parallel multiplication, which allows to perform  $\Omega(n)$  multiplications in parallel at the cost of  $\Omega(n^2)$  bits communication if the secret is of constant-size.

Finally, we note that there are many other techniques known that amortize on the cost of an entire secure computation. Here it is usually required that the computation is sufficiently structured, for instance that sufficiently many multiplications can be computed at the same point in time. For specific functions, one can additionally apply specialized optimizations that allow to reduce the amount of communication even further.

## 9.4 Formal Security Definitions for MPC

Although we do not discuss the security of the resulting protocols in detail here, we would like to point out that the security of multi-party computation can be formalized in the model of *Universal Composability* (UC) [11]. The idea behind the UC model is that any protocol is potentially used as part of a larger protocol and should remain secure regardless of the manner in which the protocol is invoked. An important advantage of this approach is that it allows to improve on previous security definitions

that are specified by a list of requirements on the protocol. This previous approach had as a strong disadvantage that the list of requirements often turned out to be incomplete once the protocols started to be used in new and unforeseen ways.

In the UC model, the functionality of a protocol is described in terms of an *ideal world* functionality. This ideal world functionality describes what we would want to achieve if we could build an ideal box that receives the inputs from the players, returns the appropriate computed outputs to the players and does nothing else that can be observed on the outside of the box.

Furthermore, to emulate the behavior of an adversary for the real-world protocol, the adversary in the ideal world communicates through a simulator that only has access to the data held by the players that are corrupted by the adversary. The idea is to construct the simulator in such a way that the adversary cannot distinguish whether he is attacking a real protocol in the real world or communicating with the simulator in the ideal world, where in the latter situation the adversary by assumption is unable to disturb the functionality of the ideal box in any way. By the properties of the UC model it then follows that the adversary is unable to do better in the real world setting than in the ideal world setting, which implies that the protocol is secure.

What should at least intuitively be clear from this description is that it suffices to prove that the view of the adversary at any point during the protocol is independent of the views of the honest players, since this allows a simulator to make up views for the honest players and feed data computed from these views to the adversary without creating inconsistencies with the correct protocol output.<sup>1</sup>

---

<sup>1</sup>If the adversary can adaptively corrupt players during the protocol the simulator additionally needs to make sure the views of the newly corrupted players are consistent with the view of the adversary. It can be shown that this can also be achieved if the view of any static adversary is independent of the view of the honest players.

## Chapter 10

# Active-Adversary Multi-Party Computation

In Chapter 9 we have only considered passive adversaries. In the presence of an active adversary, the protocols become more involved. Typically, these protocols rely on the same basic structure used for passive adversary protocols, but additionally introduce safe-guards that allow to detect and eliminate players that deviate from the protocol specification and to simulate their intended behavior among the remaining players.

One crucial building block in such protocols is an enhanced form of secret sharing called *verifiable* secret sharing (VSS). A verifiable secret sharing protocol has the properties that the secret is fixed after the initial secret sharing phases and that, in case of an honest dealer distributing the secret, the correct secret is guaranteed to be reconstructed when all players pool their shares in the secret. It is known how to construct such verifiable secret sharing protocols based on strongly multiplicative linear secret sharing schemes, which can be seen as the first step in building protocols secure against an active adversary. Although it is also possible to obtain VSS protocols without the use of strong multiplication, strongly multiplicative schemes provide error correction for free (see [24]) which trivializes the reconstruction of the secret from the shares.

In this chapter we sketch how to lift the known techniques of Cramer, Damgård and Maurer [23] for constructing protocols secure against an active adversary from strongly multiplicative linear secret sharing schemes to the more general context of strongly multiplicative ramp schemes.

The material in this chapter originates from unpublished joint work with Ignacio Cascudo Pueyo.

## 10.1 Verifiable Secret Sharing

In this section we describe a general procedure to verifiably secret share a vector with a ramp scheme. In the usual definition of a verifiable secret sharing (VSS), the VSS ensures that the following properties hold at the end of the sharing:

**Privacy** If the dealer is honest, the joint view of all corrupted players is statistically independent of the secret.

**Correctness** Either all honest players hold consistent shares in a value  $s$  or the dealer is disqualified. Additionally, when the dealer is not disqualified, it is guaranteed that the players can uniquely reconstruct the secret  $s$  by pooling their shares in  $s$ , even when some or all of the corrupted players provide an incorrect share.

We use this standard notion of VSS, except that we allow the secret to be a vector.

Assume that we have a strongly multiplicative ramp scheme  $\mathcal{M} = (\mathbb{F}_q, M, \psi)$ . We can then introduce a slightly modified version of the VSS protocol from [23], as follows.

In the VSS of [23] one player  $D$ , that is called the *dealer*, first randomly chooses a symmetric  $e \times e$  matrix  $R$  such that the first element in the first row is equal to the secret value. For the ramp scheme version of the VSS however, a symmetric  $e \times e$  matrix  $R$  is first selected at random such that the first  $k$  elements in the first row (and consequently the first  $k$  elements in the first column) are equal to the respective coordinates of the secret.

Next, the dealer sends every player  $p_j$  the vector  $\vec{v}_j = M_j R$ , which we refer to as the *share vector*. The first coordinate of this share vector is considered the *share* of player  $p_j$  in the ramp scheme, while the remaining coordinates allow to verify consistency of this share with the shares of the other players. After this the players execute a number of steps to actually perform the consistency verification.

These steps are exactly the same as in [23]:

1. Every pair of players  $p_i$  and  $p_j$  checks whether  $M_i \vec{v}_j^T = M_j \vec{v}_i^T$ .
2. If player  $p_j$  finds that for some player  $p_i$  the equality above does not hold he broadcasts a complaint.
3. In response to this complaint,  $D$  broadcasts the real value  $M_j \vec{v}_i^T$ .
4. If player  $p_j$  still does not agree he broadcasts an accusation against  $D$  and halts.
5.  $D$  must broadcast all the information in relation to player  $p_j$ . The rest of the players can check if this information is consistent with theirs and there can be new accusations against the dealer.

6. If the set of players that accuse  $D$  is in the adversary structure then the VSS is accepted. In this case every player that accused the dealer must use the information received in step 5. If, on the contrary, the set of players that accuse the dealer is not contained in the adversary structure then the dealer is disqualified.

It remains to argue that this VSS protocol is still secure when applied to ramp schemes. We sketch the proofs for correctness and privacy, considering the cases where the dealer is honest and where the dealer is dishonest.

**Honest dealer:** In this case, no honest player can ever accuse the dealer, so the set of players that accuse the dealer always consists of a subset of the adversary structure and hence the VSS is accepted, and correctness is ensured.

With regard to the privacy, note that steps 2–6 do not reveal any new information to the adversary, so we just need to argue that what the adversary receives in step 1 does not give him any further information about the secret vector. Let  $B$  be an element of the adversary structure. It is sufficient to show that there are symmetric matrices  $S_1, \dots, S_k$  such that the  $i^{\text{th}}$  element in the first row of  $S_i$  is one and the remaining  $k - 1$  first elements of this row are zeros, and such that  $M_B S_i = 0$ . If this holds, then for any secret vectors  $(s_1, \dots, s_k)$  and  $(s'_1, \dots, s'_k)$  and a symmetric matrix  $R_1$  such that its first  $k$  coordinates of its first row are  $(s_1, \dots, s_k)$ , we can compute a matrix  $R_2 = R_1 + \sum_{i=1}^k (s'_i - s_i) S_i$  such that the first  $k$  coordinates of the first row of  $R_2$  are  $(s'_1, \dots, s'_k)$  and  $M_B R_1 = M_B R_2$ . Hence every secret vector in  $\mathbb{F}_q^k$  is equally probable from the point of view of the players in the set  $B$ .

Due to the standard properties of the underlying ramp scheme there exists a  $\vec{v}_i$  such that  $M_B \vec{v}_i = 0$  and the first and  $i^{\text{th}}$  ( $1 < i \leq k$ ) coordinates of  $\vec{v}_i$  are one, while the rest of the first  $k$  coordinates are zero. Now we can take  $S_1 = \vec{v}_1 \otimes \vec{v}_1$  and  $S_i = \vec{v}_i \otimes \vec{v}_i - \vec{v}_1 \otimes \vec{v}_1$  for every  $i$  in  $\{2, \dots, k\}$ .

**Corrupt dealer:** In the case of a corrupt dealer we do not need to worry about privacy as the adversary already knows the secret vector. We only need to prove that, if the protocol is accepted, the honest players hold consistent shares at the end of the protocol. Let  $A$  be the set of honest players that accused the dealer,  $B$  be the set of corrupt players and  $C$  be the set of honest players that did not accuse the dealer during the protocol. We know that  $A$  is in the adversary structure (because the protocol was accepted), and so is  $B$ , so according to the  $R^{(3)}$  property we know that  $C$  is an accepted set. Hence, the shares of  $C$  determine uniquely the secret vector. The shares of the players in  $C$  are consistent with the new shares for the players in  $A$  that were broadcast by the dealer in step 5, which means that  $M_j \vec{v}_i^T = M_i \vec{v}_j^T$  for every player  $p_i$  in  $A$  and player  $p_j$  in  $C$ . Thus, every player  $p_j$  in  $C$  holds a share in player  $p_i$ 's share  $\vec{v}_i$ , so the first coordinate of  $\vec{v}_i$  is uniquely determined by the shares of the players in  $C$  according to the reconstruction property of the scheme.

If the VSS is accepted, at the end of the protocol all the honest players have consistent shares, and therefore the set of corrupted shares is contained in the adversary structure. In the next section we demonstrate that in this case, strong multiplicativity guarantees that honest players can efficiently and uniquely reconstruct the secret.

## 10.2 Efficient Error Correction

We now proceed by showing that strong multiplication allows to efficiently recover the secret vector even in the setting where a set of players in the adversary structure can corrupt some of their shares. We use the same kind of argument as in [24], which is in turn a generalization of the Berlekamp-Welch decoding algorithm for Reed-Solomon error correcting codes. In the following we assume that every player corresponds to a single row in the matrix corresponding to the ramp scheme, but our arguments carry over to the general case.

Throughout this section, let tensor product  $\vec{a} \otimes \vec{b}$  of any given two vectors  $\vec{a} = (a_1, a_2, \dots, a_u)$  and  $\vec{b} = (b_1, b_2, \dots, b_u)$  be the vector

$$(a_1b_1, a_1b_2, \dots, a_1b_u, \dots, a_ub_1, \dots, a_ub_u).$$

Let  $\widehat{M}$  be the matrix with  $n$  rows, where the  $i^{\text{th}}$  row equals  $M_i \otimes M_i$ . Given vectors  $\vec{s}$  and  $\vec{t}$ , let  $\vec{s} * \vec{t}$  denote the component-wise product of  $\vec{s}$  and  $\vec{t}$ . Suppose we have shares  $\vec{a} = (a_1, \dots, a_n)$  for a secret vector  $\vec{s} = (s_1, \dots, s_k)$ , so  $\vec{a} = M\vec{x}$ , where  $\vec{x}$  is a vector of the form  $(\vec{s}, \vec{\rho})$ .

Now, assume that the players in a subset  $A \in \mathcal{A}$  provide incorrect shares in the reconstruction phase. So the players actually obtain  $\vec{c} = \vec{a} + \vec{e}$ , where  $\vec{e}$  is some error vector for which the set of non-zero positions is contained in  $A$ . The players can proceed in the following way. They first find a solution  $(\vec{\gamma}, \vec{y})$  to the system of equations  $\{\widehat{M}\vec{\gamma} = \vec{c} * (M\vec{y}), y_1 = 1\}$ . The first equation can be rewritten as  $\widehat{M}(\vec{\gamma} - \vec{y} \otimes \vec{x}) = \vec{e} * (M\vec{y})$ .

The properties of the ramp scheme ensure that we can find some solution to the system of equations above. In fact, we know that there exists a vector  $\vec{z}$  with  $z_1 = 1$  and  $M_A\vec{z} = 0$ . Then  $(\vec{z} \otimes \vec{x}, \vec{z})$  is a solution for any such vector  $\vec{z}$ , as one can easily see that  $\vec{e} * (M\vec{z}) = 0$ .

Now, given any solution  $(\vec{\gamma}, \vec{y})$  to the system of equations above, consider  $\widehat{M}(\vec{\gamma} - \vec{y} \otimes \vec{x}) = (t_1, \dots, t_n)$ . Then  $t_i = 0$  for every  $i \in \overline{A}$ , because the corresponding coordinates in  $\vec{e}$  are zero. The strong multiplication property implies that the first  $k$  coordinates of  $\vec{\gamma} - \vec{y} \otimes \vec{x}$  can be written as linear combinations of all of the  $t_i$  such that  $i \in \overline{A}$  and hence these coordinates are zero. Thus, the first coordinates of  $\vec{\gamma}$  are the same as those of  $\vec{y} \otimes \vec{x}$ , which are  $(s_1, \dots, s_k)$ .

## 10.3 Active-Adversary Secure MPC

In this section we use the techniques from Chapter 9, Section 10.1 and Section 10.2 to build the protocols for multi-party computation secure against an active adversary based on strongly multiplicative ramp schemes. This generalizes the protocols given in [23] for strongly multiplicative linear secret sharing schemes.

### Active case

First, beside the VSS protocol from Section 10.1 that operates on vectors, we require a method for players to VSS single values in  $\mathbb{F}_q$  based on the underlying ramp scheme. In the VSS protocol of [23] there exist two levels of secret sharing, one for the secret and one where shares are generated for every share in the secret, and the extra VSS variant we describe here is required to mimic the latter level. Additionally we will use the new variant to perform secure multiplication.

Consider a non-zero homomorphism  $\psi : \mathbb{F}_q \rightarrow \mathbb{F}_q^k$ . The protocols below can be made to work with any such  $\psi$ , but to simplify the presentation we assume here that  $\psi$  is such that for any  $r \in \mathbb{F}_q$  we have that the first coefficient of  $\psi(r)$  equals  $r$ . To VSS an element  $r \in \mathbb{F}_q$ , a player  $p_i$  actually performs the VSS for the vector  $\psi(r)$  as described in Section 10.1. We abstractly refer to the share vectors generated while player  $p_i$  applies the VSS for the element  $r$  by  $[r]_i$ .

We follow the approach in [23], where two protocols called Commitment Transfer Protocol (CTP) and Commitment Sharing Protocol (CSP) are used. The CTP protocol allows two players  $p_i$  and  $p_j$  to transform a VSS created by player  $p_i$  into a VSS created by player  $p_j$  (i.e., the randomness and secret get “transferred” to player  $p_j$ ). The CSP protocol on the other hand allows to convert a VSS of a value  $r$  into a set of VSSes for shares  $r_1, r_2, \dots, r_n$  for  $r$ . We need to adapt these two protocols for them to work for ramp schemes, but the adaptation for the CTP protocol is trivial and therefore omitted here. We therefore proceed to describe the adapted CSP protocol:

Player  $p_i$  wants to transform a VSS he created for an element  $r \in \mathbb{F}_q$ , again denoted by  $[r]_i$ , into a set of VSS distributions  $[r_1]_1, \dots, [r_n]_n$ , where  $r_1, \dots, r_n$  are shares of  $\psi(r)$  in our ramp scheme. First note that, using the linearity of the VSS, players can already locally compute share vectors for every coordinate of  $\psi(r)$  from the share vectors corresponding to  $[r]_i$ , as every coordinate of  $\psi(r)$  is  $\lambda r$  for some constant  $\lambda$ . Additionally, player  $p_i$  performs a VSS for each entry of the random vector  $\vec{\rho}$ , where  $M(\psi(r), \vec{\rho}) = (r_1, \dots, r_n)$ . Using the distributed share vectors all players can now locally mimic the computation of  $(r_1, \dots, r_n)$ , resulting in share vectors in the values  $r_1, \dots, r_n$ . Since the values  $r_1, \dots, r_n$  are at this point still only known by player  $p_i$ , these share vectors correspond to VSS distributions  $[r_1]_i, [r_2]_i, \dots, [r_n]_i$ . Finally, for every player  $p_j$ , players  $p_i$  and  $p_j$  use the CTP protocol to convert  $[r_j]_i$  into  $[r_j]_j$ .

We are now ready to describe the multi-party computation protocol secure against an active adversary.

## Initialization phase

At the beginning of the protocol, every player holds a secret vector of size  $k$ . The goal of this phase is to create share vectors for these vectors according to the VSS.

Player  $p_i$ , who has the secret vector  $\vec{s}$ , proceeds in the following way:

1. Player  $p_i$  chooses a vector  $\vec{\rho}$  uniformly at random, which he uses to share  $\vec{s}$  with the strongly multiplicative ramp scheme. Next he performs VSS separately for each entry of  $\vec{s}$  and  $\vec{\rho}$ .
2. Players can compute  $[a_j]_i$ , share vectors for the shares  $a_j$ , as they are linear combinations of the entries of  $\vec{s}$  and  $\vec{\rho}$ .
3. Players  $p_i$  and  $p_j$  execute the CTP protocol to transform  $[a_j]_i$  into  $[a_j]_j$ . After this step player  $p_j$  knows the value  $a_j$ .

After this initialization phase every player obtained a share in the secret vector of every player and share vectors for the shares that every player obtained. To analyze the communication complexity of this protocol assume that  $k = O(n)$  and that the size of the random vector  $\vec{\rho}$  is  $O(n)$ . Every player performs VSS for each entry of a vector of length  $O(n)$  and every VSS amounts to sending every player a vector of size  $O(n)$ , so the complexity of every VSS is  $O(n^2)$ . As there are  $n$  players the total complexity of this phase is  $O(n^4)$ .

## Secure multiplication

Using the linearity of ramp schemes and the VSS scheme, secure addition and multiplication with a constant are achieved similar to the passive case and can be performed locally by the players without interaction. We therefore focus on the more involved and interesting operation, which is secure multiplication.

Let  $\vec{s}$  and  $\vec{t}$  be two secret vectors. Every player has a VSS'ed share for each vector and wants to compute a VSS'ed share to  $\vec{s} \odot \vec{t}$ . Let us consider a player  $p_i$  and a VSS distribution  $[a]_i$  for his share  $a$  to  $\vec{s}$  and VSS distribution  $[b]_i$  for his share  $b$  to  $\vec{t}$ . Player  $p_i$  performs the following actions:

1. Player  $p_i$  applies the CSP protocol to  $[a]_i$  and  $[b]_i$ , thus creating VSS distributions for shares  $\{a_j\}_{j=1}^n$  and  $\{b_j\}_{j=1}^n$  for  $a$  and  $b$  respectively for every player, that is, every player  $p_j$  obtains distributions  $[a_j]_j$  and  $[b_j]_j$  and learns  $a_j$  and  $b_j$ . In this process player  $p_i$  uses some random vectors  $\vec{\rho}_a$  and  $\vec{\rho}_b$  of length  $e - k$ .

2. Player  $p_i$  additionally performs VSS for  $ab$ , resulting in  $[ab]_i$ .
3. Player  $p_i$  computes a vector  $\vec{\rho}$  for which it holds that

$$\widehat{M}(ab, \vec{\rho}) = M(\psi(a), \vec{\rho}_a) * M(\psi(b), \vec{\rho}_b)$$

and then performs VSS for each entry of  $\vec{\rho}$ .

4. Let  $\widehat{M}(ab, \vec{\rho}) = ((ab)_1, (ab)_2, \dots, (ab)_n)$ . The players locally compute share vectors corresponding with  $[(ab)_j]_i$  using  $[ab]_i$ , the share vectors corresponding with the entries of  $\vec{\rho}$  and the fact that the VSS scheme and the mapping  $\psi$  have homomorphic properties.
5. All players send their share in  $[(ab)_j]_i$  to player  $p_j$ , who verifies that  $(ab)_j$  is equal to  $a_j b_j$ . In case it is not, player  $p_j$  complains and broadcasts the values  $a_j$  and  $b_j$ , together with the randomness used to create the distributions  $[a_j]_j$  and  $[b_j]_j$ . Player  $p_i$  must then do the same for the distribution  $[(ab)_j]_i$ . If  $a_j b_j \neq (ab)_j$  player  $p_i$  is disqualified. All of the values that were shared via VSS by him are publicly determined and the players restart the protocol again simulating this values.
6. Now player  $p_i$  performs VSS for (each entry of) a different random vector  $\vec{\sigma}$  of length  $e - k$ .
7. Based on their share vectors corresponding with  $[ab]_i$ , the players compute without interaction  $[\lambda_i^{(1)} ab]_i, [\lambda_i^{(2)} ab]_i, \dots, [\lambda_i^{(k)} ab]_i$ , where  $\lambda_i^{(1)}, \lambda_i^{(2)}, \dots, \lambda_i^{(k)}$  are the constant values from Definition 36.
8. Every player  $p_j$  can locally compute share vectors corresponding to shares for the vector  $\vec{\lambda}_i ab$  using  $[\vec{\sigma}]_i$  and  $[\lambda_i^{(1)} ab]_i, [\lambda_i^{(2)} ab]_i, \dots, [\lambda_i^{(k)} ab]_i$ . We refer to these VSS distributions by  $[x_{ji}]_i$ .
9. Players  $p_i$  and  $p_j$  perform the CTP protocol, so that they construct  $[x_{ji}]_j$ , and player  $p_j$  learns  $\psi(x_{ji})$  and hence  $x_{ji}$ .
10. Each player  $p_j$  sums his shares  $x_{ji}$ . The result  $\sum_i x_{ji}$  is a share for the vector  $s \odot t$ . Similarly, VSS distributions  $[\sum_i x_{ji}]_j$  are computed.

Since all the values are distributed via VSS the only way player  $p_i$  can cheat in the above protocol is by performing VSS on a value  $c \neq ab$  in step 2. We next argue that in this case he would be disqualified in step 5. Indeed, we know that if he is not disqualified  $c_j$  must equal  $a_j b_j$  for every honest player, that is for a set of players which is in the complement of the adversary structure. Therefore, by the strong multiplication property, these shares determine the secret and thus  $c = ab$ .

Thus, steps 1 – 5 allow players to check that player  $p_i$  has really performed VSS on the value  $ab$  and can be seen as a translation for the ramp schemes context of the CMP protocol in [23]. In steps 6 – 9, each player  $p_j$  constructs a VSS for a share in the vector  $\vec{\lambda}_i ab$ , where  $a$  and  $b$  are the shares of player  $p_i$  for  $s$  and  $t$ . Due to the multiplication property, the sum of these vectors gives  $s \odot t$ . So after step 10, every player  $p_j$  has a VSS distribution for a share in  $s \odot t$ .

The most expensive step in this protocol with respect to the communication complexity is step 3. In this step player  $p_i$  must perform VSS for each entry of a random vector  $\vec{\rho}$  which is of length  $O(n^2)$ . As we know that the communication complexity of a single VSS is  $O(n^2)$  and each of the  $n$  players must perform VSS on  $O(n^2)$  elements of  $\mathbb{F}_q$  this yields a total communication complexity of  $O(n^5)$ .

## 10.4 Active-Adversary MPC from AG Schemes

It is worth noting that for specific ramp schemes one can often construct secure protocols with lower communication complexity. As an example, we next sketch how we can construct specific protocols secure against an active adversary for the strongly multiplicative algebraic-geometric ramp schemes detailed in Chapter 8. These protocols require the communication of  $O(n^3)$  field elements while operating on vectors consisting of  $k$  elements in  $\mathbb{F}_q$ .

More specifically, we sketch how to obtain more efficient multi-party computation protocols secure against an active adversary based on the algebraic-geometric ramp schemes from Section 8.2, provided that  $n \geq 4t + 4g + 2k - 1$ , where the adversary is an active  $t$ -adversary and the secret vectors are of length  $k$  over  $\mathbb{F}_q$ . The relaxation of the parameters with regard to the general approach from the previous section has to do with the idea that when every player multiplies two shares in two different vectors, but based on the same algebraic-geometric ramp scheme using rational functions in  $L(G)$ , the resulting value can be seen as a share in the product of the secret vectors in a new ramp scheme using rational function in  $L(2G)$ . We require this new ramp scheme to allow for efficient error correction, which is the case for the parameters we assume. In particular, these parameters allow to perform VSS based on the new ramp scheme using rational function in  $L(2G)$ , which will be used heavily to perform verification checks on multiplied secret vectors.

Since during the secure computation the only operation requiring communication is multiplication, and all other operations are essentially implemented around this operation, we mainly focus on secure multiplication and the ideas involved in improving its efficiency.

**Multiplication**

We first note that the general structure of any multiplication subprotocol in the active case is essentially the same as in the passive case. First, every player  $p_i$ :

1. Reshapes the product  $a_i b_i$  of his shares  $a_i$  and  $b_i$  in the VSS of the secret vectors that are to be multiplied.
2. Reshapes his contribution  $\vec{\lambda}_i a_i b_i = (\lambda_i^{(1)} a_i b_i, \lambda_i^{(2)} a_i b_i, \dots, \lambda_i^{(k)} a_i b_i)$  in the output of the multiplication.

After this the players can add up their shares in the contributions  $\vec{\lambda}_i a_i b_i$  of the players to obtain shares in the product  $\vec{s} \odot \vec{t} = \sum_{i=1}^n \vec{\lambda}_i a_i b_i$ .

The real issue concerns the fact that players need to be able to verify that every player  $p_i$  reshapes the correct value  $a_i b_i$  and subsequently correctly reshapes the vector  $\vec{\lambda}_i a_i b_i$ . For our solution, we introduce methods that allow to verify that two secret shared vectors contain respectively the same first coefficient and all the same coefficients.

The basic idea is as follows. As mentioned earlier, when the players hold shares in two secret vectors distributed using rational functions in an appropriate Riemann-Roch space  $L(G)$  (see Section 8.2 for more details), the local products of these shares can again be seen as shares in the product of the secret vectors in a new ramp scheme based on rational functions in  $L(2G)$ , which is a space that strictly contains the space  $L(G)$  used for the original ramp scheme. By an extension of this idea, and using the linearity of the VSS, one can actually obtain share vectors corresponding to a VSS based on the new ramp scheme from the share vectors in the original VSS. However, to simplify the discussion we mostly omit the extra level introduced by the VSS from here on.

The idea that allows to verify the first resharing step is now to let player  $p_i$  reshare his local product share  $a_i b_i$  using the original VSS based on  $L(G)$ , which can be seen as embedded in the space  $L(2G)$ , so that when this is done correctly the players have a set of shares in  $a_i b_i$  in two seemingly different schemes. If this is done in such a way that the secret vectors encoding  $a_i b_i$  have some similarities, for instance both contain  $a_i b_i$  in their first position, players can identify whether both secret vectors correspond to the same value by looking at their difference. By setting up the basis for  $L(G)$  and  $L(2G)$  correctly this can be performed by letting the players take the local difference of their shares, which leads to shares in the new scheme over  $L(2G)$ , and publicly determine the corresponding secret vector. Although we do not add the details here, we note that to preserve privacy some random padding needs to be added to this procedure.

In order to verify the second resharing, the players first themselves compute shares in the vector  $\vec{\lambda}_i a_i b_i$  in the new ramp scheme over  $L(2G)$  using the new shares in the

value  $a_i b_i$  in the original ramp scheme and then verify whether player  $p_i$  secret shared the correct vector using the comparison technique on secret vectors just described for the first verification.

In order to make this computation for the players possible, they require the value  $a_i b_i$  to be reshared using very specific homomorphisms  $\psi : \mathbb{F}_q \rightarrow \mathbb{F}_q^k$  (see Section 10.3), depending on the algebraic-geometric ramp scheme that is considered:

- *Parallel Multiplication:* The map  $\psi$  is defined by  $c \mapsto (c, c, \dots, c)$ .
- *Extension Field Multiplication:* The map  $\psi$  is defined by  $c \mapsto (c, 0, \dots, 0)$ .

These maps are chosen in such a way that

$$\psi(a_i b_i) \odot (\lambda_1^{(i)}, \lambda_2^{(i)}, \dots, \lambda_k^{(i)}) = (\lambda_1^{(i)} a_i b_i, \lambda_2^{(i)} a_i b_i, \dots, \lambda_k^{(i)} a_i b_i),$$

for the relevant multiplication operation  $\odot$ , which is straightforward to verify.

To ensure that  $a_i b_i$  is reshared using the correct map  $\psi$  we additionally modify the VSS protocol from Section 10.1 in various ways to allow to impose structural restrictions on the secret vector, i.e., guarantee the presence of zeroes in prespecified locations or repetition of some value in all the positions. For additional details, the interested reader is referred to [16].

# Bibliography

- [1] S. Agarwal, R. Cramer, and R. de Haan. Asymptotically Optimal Two-Round Perfectly Secure Message Transmission. In *Advances in Cryptology: CRYPTO '06*, volume 4117 of *Lecture Notes in Computer Science*, pages 389–401. Springer-Verlag, 2006.
- [2] J. Bar-Ilan and D. Beaver. Non-cryptographic fault-tolerant computing in a constant number of rounds. In *8th ACM Symposium Annual on Principles of Distributed Computing*, pages 201–209, 1989.
- [3] D. Beaver. Minimal latency secure function evaluation. In *Advances in Cryptology: EUROCRYPT '00*, volume 1807 of *Lecture Notes in Computer Science*, pages 335–350. Springer-Verlag, 2000.
- [4] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, pages 1–10. Springer-Verlag, 1988.
- [5] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer. Generalized privacy amplification. *IEEE Transactions on Information Theory*, 41(6-2):1915–1923, 1995.
- [6] C. H. Bennett, G. Brassard, and J.-M. Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17:210–229, 1988.
- [7] C.H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179, 1984.
- [8] E. Berlekamp and L. Welch. Error correction of algebraic block codes. US Patent 4,633,470.

- [9] E.F. Brickell. Some Ideal Secret Sharing Schemes. In *Advances in Cryptology: EUROCRYPT '89*, volume 434 of *Lecture Notes in Computer Science*, pages 468–475. Springer-Verlag, 1990.
- [10] C. Cachin and U. Maurer. Unconditional Security Against Memory-Bounded Adversaries. In *Advances in Cryptology: CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 292–306. Springer-Verlag, 1997.
- [11] R. Canetti. Security and Composition of Multiparty Cryptographic Protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
- [12] R. Canetti, U. Feige, O. Goldreich, and M. Naor. Adaptively Secure Multi-Party Computation. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96)*, pages 639–648. Springer-Verlag, 1996.
- [13] R.M. Capocelli, A. De Santis, L. Gargano, and U. Vaccaro. On the Size of Shares for Secret Sharing Schemes. In *Advances in Cryptology: CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 101–113. Springer-Verlag, 1991.
- [14] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC '88)*, pages 11–19. Springer-Verlag, 1988.
- [15] H. Chen and R. Cramer. Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computation over Small Fields. In *Proceedings of 26th Annual IACR CRYPTO*, volume 4117, pages 521–536, Santa Barbara, Ca., USA, August 2006. Springer Verlag LNCS.
- [16] H. Chen, R. Cramer, R. de Haan, and I. Cascudo Pueyo. Strongly Multiplicative Ramp schemes from High Degree Rational Points on Curves. In *Proceedings of EUROCRYPT 2008*, volume 4965, pages 451–470. Springer Verlag LNCS, 2008.
- [17] H. Chen, R. Cramer, S. Goldwasser, R. de Haan, and V. Vaikuntanathan. Secure Computation from Random Error Correcting Codes. In *Proceedings of 26th Annual IACR EUROCRYPT 2007*, volume 4515, pages 291–310. Springer Verlag LNCS, 2007.
- [18] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.
- [19] R. Cramer. Perfectly Secure Message Transmission V1.1, September 2006. Lecture notes Dutch National Master's Program on Number Theory and Cryptology.

- [20] R. Cramer and I. Damgård. Linear Zero-Knowledge - A Note on Efficient Zero-Knowledge Proofs and Arguments. *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pages 436–445, 1997.
- [21] R. Cramer, I. Damgård, and R. de Haan. Atomic Secure Multi-Party Multiplication with Low Communication. In *Proceedings of EUROCRYPT 2007*, volume 4515, pages 329–346. Springer Verlag LNCS, 2007.
- [22] R. Cramer, I. Damgård, and S. Dziembowski. On the Complexity of Verifiable Secret Sharing and Multi-Party Computation. In *Proceedings of the 32th Annual ACM Symposium on Theory of Computing (STOC '00)*, pages 325–334. Springer-Verlag, 2000.
- [23] R. Cramer, I. Damgaard, and U. Maurer. General Secure Multi-Party Computation from any Linear Secret-Sharing Scheme. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer-Verlag, May 2000.
- [24] R. Cramer, V. Daza, I. Gracia, J. Jiménez Urroz, G. Leander, J. Martí-Farré, and C. Padró. On codes, matroids and secure multi-party computation from linear secret sharing schemes. In *Proceedings of CRYPTO 2005*, volume 3621 of *LNCS*, pages 327–343. Springer-Verlag, 2005.
- [25] R. Cramer and S. Fehr. Optimal Black-Box Secret Sharing over Arbitrary Abelian Group. In *Advances in Cryptology - CRYPTO '02*, volume 2442 of *Lecture Notes in Computer Science*, pages 272–287. Springer-Verlag, 2002.
- [26] I. Damgård, S. Fehr, L. Salvail, and C. Schaffner. Cryptography in the Bounded Quantum-Storage Model. *SIAM Journal on Computing*, 37(6):1865–1890, 2008.
- [27] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1967.
- [28] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly Secure Message Transmission. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS '90)*, pages 36–45. IEEE Computer Society Press, 1990.
- [29] D. Dolev, C. Dwork, O. Waarts, and M. Yung. Perfectly Secure Message Transmission. *Journal of the ACM*, 40(1):17–47, January 1993.
- [30] I.M. Duursma. Algebraic decoding using special divisors. *IEEE Transactions on Information Theory*, 39(2):694–698, 1993.

- [31] Bogetoft et al. Multi-Party Computation Goes Live. Cryptology ePrint Archive, 2008. <http://eprint.iacr.org>.
- [32] U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation. In *26th ACM Symposium Annual on Principles of Distributed Computing*, pages 554–563. ACM Press, 1994.
- [33] M. Fitzi, M. Franklin, J. Garay, and S. Harsha Vardhan. Towards Optimal and Efficient Perfectly Secure Message Transmission. In *Theory of Cryptography Conference: TCC '07*, volume 4392 of *Lecture Notes in Computer Science*. Springer-Verlag, 2007.
- [34] M. Franklin and M. Yung. Communication complexity of secure computation. In *Proceedings of STOC 1992*, pages 699–710. ACM Press, 1992.
- [35] P. Gaborit and A. Otmani. Experimental constructions of self-dual codes. Manuscript. Available from [http://www.unilim.fr/pages\\_perso/philippe.gaborit/SD/](http://www.unilim.fr/pages_perso/philippe.gaborit/SD/), 2002.
- [36] A. Garcia and H. Stichtenoth. On the asymptotic behavior of some towers of function fields over finite fields. *J. Number Theory*, 61:248–273, 1996.
- [37] E.N. Gilbert. A comparison of signalling alphabets. *Bell Syst. Tech. Journal*, 31:504–522, 1952.
- [38] O. Goldreich, S. Micali, and A. Wigderson. How to Play Any Mental Game or a Completeness Theorem for Protocols with Honest Majority. In *Proceedings of the 6th ACM Symposium on Principles of Distributed Computing (PODC '87)*, pages 218–229. ACM, 1987.
- [39] O. Goldreich, S. Micali, and A. Wigderson. How to Solve any Protocol Problem. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC '87)*. Springer-Verlag, 1987.
- [40] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract). *Proceedings of the 17th Annual ACM Symposium on Theory of Computing (STOC '85)*, pages 291–304, 1985.
- [41] V.D. Goppa. A new class of linear error-correcting codes. *Problemy Peredachi Informatsii*, 6(3):207–212, 1970.
- [42] M. Hirt and U. Maurer. Player Simulation and General Adversary Structures in Perfect Multiparty Computation. *Journal of Cryptology*, 13(1):31–60, April 2000. Extended abstract in *Proc. 16th of ACM PODC '97*.

- [43] M. Hirt, U. Maurer, and B. Przydatek. Efficient Secure Multi-Party Computation. In *Advances in Cryptology: ASIACRYPT '00*, Lecture Notes in Computer Science, pages 143–161. Springer-Verlag, 2000.
- [44] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
- [45] Y. Ihara. Some remarks on the number of rational points of algebraic curves over finite fields. *J. Fac. Sci. Univ. Tokyo Sect. IA Math.*, 28:721–724, 1981.
- [46] Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. In *Proc. 5th Israel Symposium on Theoretical Comp. Sc. ISTCS*, pages 174–183, 1997.
- [47] Y. Ishai and E. Kushilevitz. Randomizing polynomials: A new paradigm for round-efficient secure computation. In *41th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 2000.
- [48] Y. Ishai, E. Kushilevitz, R. Ostrovsky, and A. Sahai. Zero-knowledge from secure multiparty computation. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC '07)*, pages 21–30. Springer-Verlag, 2007.
- [49] M. Karchmer and A. Wigderson. On Span Programs. In *Proceedings of the 8th Annual Symposium on Structure in Complexity Theory*, pages 102–111, 1993.
- [50] E.D. Karnin, J.W. Greene, and M.E. Hellman. On Secret Sharing Systems. *IEEE. Transactions on Information Theory*, IT-29(1):35–41, 1983.
- [51] K. Kurosawa and K. Suzuki. Truly Efficient 2-Round Perfectly Secure Message Transmission Scheme. In *Advances in Cryptology: EUROCRYPT '08*, volume 4965 of *Lecture Notes in Computer Science*, pages 324–340. Springer-Verlag, 2008.
- [52] L. Lamport, R. Shostak, and M. Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, July 1982.
- [53] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier, 2006.
- [54] F.J. MacWilliams, N.J.A. Sloane, and J.G. Thompson. Good self-dual codes exist. *Discrete Math.*, 3:153–162, 1972.
- [55] J.L. Massey. Shift-register synthesis and BCH decoding. *IEEE Trans. Inform. Theory*, IT-15:122–127, 1969.

- [56] J.L. Massey. Minimal codewords and secret sharing. In *Proceedings of the 6-th Joint Swedish-Russian Workshop on Information Theory*, pages 269–279, Molle, Sweden, August 1993.
- [57] J.L. Massey. Some applications of coding theory in cryptography. *Codes and Ciphers: Cryptography and Coding IV*, pages 33–47, 1995.
- [58] U. Maurer. Secret Key Agreement by Public Discussion. *IEEE Transaction on Information Theory*, 39(3):733–742, May 1993.
- [59] L.H. Ozarow and A.D. Wyner. “Wire-tap-channel II”. *AT&T Bell Labs Tech. J.*, 63:2135–2157, 1984.
- [60] V.S. Pless, editor. *Handbook of Coding Theory*. Elsevier, 1998.
- [61] E.M. Rains and N.J.A. Sloane. Self-Dual Codes. A long survey article written for the Handbook of Coding Theory. Available from <http://www.research.att.com/~njas/>, 1998.
- [62] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.
- [63] S. Sakata, J. Justesen, Y. Madelung, H.E. Jensen, and T. Høholdt. Fast decoding of algebraic-geometric codes up to the designed minimum distance. *IEEE Transactions on Information Theory*, 41(6):1672–1677, 1995.
- [64] H. Sayeed and H. Abu-Amara. Efficient Perfectly Secure Message Transmission in Synchronous Networks. *Information and Communication*, 126(1):53–61, 1996.
- [65] J.-P. Serre. Rational points on curves over finite fields. Notes by F. Gouvea of lectures at Harvard University, 1985.
- [66] A. Shamir. How to Share a Secret. *Communications of the ACM*, 22:612–613, 1979.
- [67] C.E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27:379–423 & 623–656, 1948.
- [68] V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2008.
- [69] J.H. Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer-Verlag, 1986.

- [70] K. Srinathan, A. Narayanan, and C. Pandu Rangan. Optimal Perfectly Secure Message Transmission. In *Advances in Cryptology: CRYPTO '04*, volume 3152 of *Lecture Notes in Computer Science*, pages 545–561. Springer-Verlag, 2004.
- [71] K. Srinathan, N.R. Prasad, and C. Pandu Rangan. On the Optimal Communication Complexity of Multiphase Protocols for Perfect Communication. In *IEEE Symposium on Security and Privacy*, pages 311–320. IEEE Computer Society, 2007.
- [72] H. Stichtenoth. *Algebraic Function Fields and Codes*. Springer-Verlag, 1993.
- [73] D.R. Stinson. An Explication of Secret Sharing Schemes. *Des. Codes Cryptography*, 2(4):357–390, 1992.
- [74] J.G. Thompson. Weighted averages associated to some codes. *Scripta Math.*, 29:449–452, 1973.
- [75] M.A. Tsfasman and S.G. Vlăduț. *Algebraic-Geometric Codes*. Kluwer, 1991.
- [76] M.A. Tsfasman, S.G. Vlăduț, and T. Zink. Modular curves, Shimura curves and Goppa codes better than the Varshamov-Gilbert bound. *Math. Nachr.*, 109:21–28, 1982.
- [77] C. Xing. Nonlinear codes from algebraic curves improving the Tsfasman-Vladut-Zink bound. *IEEE Transactions on Information Theory*, 49(7):1653–1657, 2003.
- [78] A.C. Yao. Protocols for Secure Computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '82)*, pages 160–164. IEEE, 1982.



# Nederlandse Samenvatting

Dit proefschrift behandelt nieuwe resultaten in twee gebieden binnen de cryptografie; het veilig verzenden van een boodschap tussen twee partijen en het veilig uitvoeren van een berekening op invoer van meerdere partijen. We omschrijven eerst de context voor het veilig verzenden van een boodschap tussen twee partijen. Dit is een relatief oud probleem, waar de focus zowel ligt op het voorkomen dat iemand die de boodschap onderschept hem kan ontcijferen als het voorkomen dat iemand een boodschap onderweg kan vervangen door een andere boodschap zonder dat de ontvanger dit opmerkt.

De standaard opstelling voor dit probleem omvat twee partijen, een verzender en een ontvanger, die verbonden zijn via een onveilig kanaal waarop een derde partij alleen af luistert of in het ergste geval zelfs data kan vervangen of blokkeren. Na een rijke geschiedenis aan resultaten zijn er momenteel technieken om af luisteren en/of het ongemerkt vervangen van data tegen te gaan, hoewel de meeste technieken gebruik maken van een complexiteitsaanname op een of ander momenteel moeilijk op te lossen wiskundig probleem en er vaak een minuscule kleine foutkans aanwezig is. Aangezien er in de standaardopstelling sprake is van een enkel communicatiekanaal is het onmogelijk het blokkeren van boodschappen tegen te gaan wanneer het kanaal deze functionaliteit niet al aanbiedt.

In dit proefschrift beschouwen we een recenter geïntroduceerd model, waar de verzender en ontvanger verbonden zijn via meerdere communicatiekanalen. Als een derde partij in dit model niet teveel kanalen tegelijk kan manipuleren is het mogelijk volledig veilige communicatie te garanderen, dat wil zeggen zonder foutkans of informatielekage en zonder het gebruik van een wiskundige complexiteitsaanname. In dit proefschrift geven we een overzicht van alle volledig veilige bekende resultaten voor dit model en bepalen we tevens de precieze best mogelijke communicatiecomplexiteit die hier behaald kan worden onder de zwakst mogelijke aanname dat slechts een minderheid van de kanalen tegelijk gecorrumpeerd kan worden door een derde partij.

Het tweede onderwerp dat we aansnijden betreft het veilig uitvoeren van een berekening op invoer van meerdere partijen in een situatie waar een onbekend deel

van deze partijen onder de controle staat van een kwaadaardige entiteit. Afhankelijk van de situatie is deze entiteit alleen in staat de communicatie en lokale data van de partijen onder controle in te lezen danwel in staat de acties van deze partijen geheel te bepalen. In beide situaties is het vereist dat de correcte uitvoer van de berekening door alle partijen bepaald wordt zonder dat extra informatie betreffende de invoer van de “eerlijke” partijen naar de kwaadaardige entiteit lekt die niet toch al bepaald kon worden aan de hand van de uitvoer en de invoerwaarden van de partijen onder zijn controle.

In het model dat wij beschouwen zijn alle partijen onderling verbonden via volledig veilige communicatiekanalen en hebben zij tevens toegang tot een functionaliteit die een partij in staat stelt een identieke boodschap tegelijk naar alle andere partijen te versturen. Aangenomen dat niet teveel partijen onder de controle van de kwaadaardige entiteit kunnen geraken zijn er oplossingen bekend in dit model die de vereisten met perfectie behalen.

In dit proefschrift introduceren we nieuwe technieken die de vereiste communicatie voor dit soort berekeningen sterk verminderen, waar we onder andere nieuwe verbanden leggen met de coderingstheorie en de algebraïsche meetkunde.

# Acknowledgements

Very few Ph.D. theses are written in isolation and this thesis is no exception. It is therefore only appropriate to mention, within the limitations imposed by Leiden University, the people that have been involved in the process.

First of all, I would like to thank Herman te Riele for mentioning the new crypto group when it was about to be set up at CWI and in particular for using one of his post-doc grants to sponsor half of my Ph.D. position. Although it is regrettable that this did not result in any scientific cooperation between us, the act is still highly appreciated.

My supervisor Ronald Cramer and I have had many interesting discussions and interactions in the last four years. These have proven invaluable to improving my skills as a researcher as well as my presentation skills and have eventually led to this thesis. In my opinion the quality of this thesis is a fair reflection of the success of our cooperation.

The first few years of my Ph.D. I ended up sharing an office with Serge Fehr, which has been a very educational experience. I think it is safe to say that my interactions with Serge have played an important role in shaping me up for the remainder of my Ph.D. track. I would like to thank Serge Fehr, Dennis Hofheinz and Eike Kiltz for several pleasant and some useful discussions that have widened my view of cryptography and increased my understanding of certain technical issues.

I especially enjoyed Ignacio Cascudo's visits to CWI. Ignacio's burning interest in mathematics and his eye for detail made it very pleasant to work with him. We had many good discussions together during these times and together worked on several problems.

Ignacio Cascudo has done some thorough initial proof-reading on the sections on secure message transmission. Moreover, Berry Schoenmakers and Hendrik Lenstra have provided some useful suggestions and comments concerning other parts of this thesis.

Tobias Baanders did an excellent job designing the cover of my thesis. Thanks to him the thesis looks a lot more lively than before and immediately gives a nice initial

impression of the topics that it covers.

Finally, I would like to thank my family, friends and in particular my lovely wife Zhan for their continuing support and confidence in me during the last four years. It has meant a lot to me.

# Curriculum Vitae

## Personal Information:

**Name:** Robbert de Haan

**Date of Birth:** Februari 21, 1980

**Place of Birth:** Amsterdam, The Netherlands

## Education:

**2004-2009:** Ph.D. student at the Centrum voor Wiskunde en Informatica, Amsterdam & the Mathematical Institute, Leiden University. Thesis: “Algebraic Techniques for Low Communication Secure Protocols”. Supervisor: Prof. dr. R. Cramer.

**1999-2004:** Master’s degree in Mathematics at the University of Amsterdam, Korteweg-de Vries Instituut. Thesis: “A fast, rigorous technique for verifying the regulator of a real quadratic field” (Research done at the Centre for Information Security and Cryptography, Calgary, Canada). Supervisors: Dr. M.J. Jacobson, Jr., prof. dr. H.C. Williams and dr. R.W. van der Waall.

**1999-2003:** Master’s degree in Computer Science at the University of Amsterdam, FNWI. Thesis: “Using ASF+SDF for the Verification of Annotated Java Programs” (Research done at the Centrum voor Wiskunde en Informatica, Amsterdam). Supervisors: Prof. dr. P. Klint and dr. F.S. de Boer.

**1998-1999:** Propedeuse Mathematics (Cum Laude) at the University of Amsterdam, FNWI.

**1998-1999:** Propedeuse Computer Science (Cum Laude) at the University of Amsterdam, FNWI.

**1992-1998:** VWO at the Sint Nicolaas Lyceum, Amsterdam.

**Publications:**

1) H. Chen, R. Cramer, R. de Haan and I. Cascudo Pueyo. *Strongly Multiplicative Ramp Schemes from High Degree Rational Points on Curves*. In: Proceedings IACR EUROCRYPT 2008, Istanbul, Turkey, Springer Verlag LNCS, vol. 4965, pp. 451–470, April 2008.

2) R. de Haan, M. J. Jacobson, Jr. and H. C. Williams. *A fast, rigorous technique for computing the regulator of a real quadratic field*. In: Mathematics of Computation, vol. 76, pp. 2139–2160, October 2007.

3) H. Chen, R. Cramer, S. Goldwasser and R. de Haan, V. Vaikuntanathan. *Secure Computation from Random Error Correcting Codes*. In: Proceedings of 26th Annual IACR EUROCRYPT, Barcelona, Spain, Springer Verlag LNCS, vol. 4515, pp. 291–310, May 2007.

4) R. Cramer, I. Damgaard and R. de Haan. *Atomic Secure Multiplication with Low Communication*. In: Proceedings of 26th Annual IACR EUROCRYPT, Barcelona, Spain, Springer Verlag LNCS, vol. 4515, pp. 329–346, May 2007.

5) S. Agarwal, R. Cramer and R. de Haan. *Asymptotically Optimal Perfectly Secure Message Transmission*. In: Proceedings of 26th Annual IACR CRYPTO, Santa Barbara, Ca., USA, Springer Verlag LNCS, vol. 4117, pp. 389–401, August 2006.