Karlsruhe Institute of Technology

Computational Arithmetic Secret Sharing and Secure Multiparty Computation

Diploma Thesis of

Alexander Koch

At the Department of Informatics,
Institute of Theoretical Informatics

and the Department of Mathematics,
Institute for Algebra and Geometry

Reviewers:	Jun.-Prof. Dr. Dennis Hofheinz PD Dr. Stefan Kühnlein
Time Period:	July 2013 – March 2014

Danksagung

Ich möchte mich an dieser Stelle ganz herzlich bei meinen beiden Betreuern Jun.-Prof. Dr. Dennis Hofheinz und PD Dr. Stefan Kühnlein für hilfreiche Diskussionen und Anmerkungen bedanken, die zum Entstehen dieser Arbeit entscheidend beigetragen haben. Die eingeräumten Freiheiten bei der Festlegung und Weiterentwicklung des Themas habe ich sehr zu schätzen gewusst. Ich danke auch Prof. Dr. Jörn Müller-Quade für die Übernahme der Zweitkorrektur.

Während meiner Studienzeit und der Anfertigung dieser Arbeit haben mich noch weitere Personen unterstützt. Hier möchte ich vor allem Oliver Thomas, Daniel Mendler und Thomas Grombein dankend erwähnen. Des Weiteren möchte ich meiner Familie danken, die mir Kraft und den nötigen Rückhalt gegeben hat. Abschließend danke ich auch der Studienstiftung des deutschen Volkes für ihre hilfreiche und vielseitige Unterstützung.

Statement of Authorship

I hereby declare that this document has been composed by myself and describes my own work, unless otherwise acknowledged in the text.

Karlsruhe, March 18, 2014.

Abstract

Secret sharing schemes allow for sharing a secret message so that it can be correctly reconstructed in the presence of enough of its shares, but with the property that nothing can be learned about its content if too few of the shares have been obtained. Homomorphic schemes exhibit the additional property that it is possible to calculate on the shares to obtain a share of the sums and products of secrets—yielding a plethora of applications, including so-called secure multiparty computation (MPC). With MPC, several parties can jointly compute an arbitrary function, without learning anything about the input values of the other parties.

To reduce the size of the generated shares in a secret sharing scheme, so-called “computational” variants have been developed which guarantee secrecy for illegitimate access to the secret only against adversaries that are bounded in their computation time by a polynomial in the input length. While these schemes are much more efficient with respect to their share size, they have the disadvantage of not being homomorphic. In this thesis, we develop a secret sharing scheme on the basis of fully homomorphic encryption, that combines the advantages of both worlds. It is therefore space efficient and allows for calculation on the shares of a secret. For this, the used encryption scheme needs to be homomorphic also with respect to ciphertexts which are encrypted under different cryptographic keys. Because of this, we include a proof of this property for an improved variant of such an encryption scheme.

The second part of the thesis starts with an introduction to the theory of algebraic function fields and algebraic geometric codes. For these, the Riemann–Roch theorem is central, and is therefore illustrated in this context. The basis of this part is the paper of Cascudo, Cramer, and Xing [CCX12b], who construct an infinite family of so-called d-arithmetic (i. e. compatible with d-fold multiplications) secret sharing schemes with good asymptotic properties. The construction of the underlying codes is achieved by infinite class field towers with the help of a group theoretic argument by Golod and Šafarevič. To ensure that these are well-suited for the purpose of secret sharing, the defining divisors are determined by so-called Riemann–Roch systems of equations. These systems are a generalization of previous methods and allow for taking the d-torsion of the zero divisor class group into account.

The so-constructed d-arithmetic secret sharing schemes are moreover used as a building block for the construction of the computational d-arithmetic scheme. Afterwards, it is described how it can be used to obtain a protocol for MPC, although it does not have the same linearity property usually found in the literature. Moreover, the connection of the notions used in the field of provable security and the ones used in algebraic coding theory, is highlighted.

Zusammenfassung

Geheimnisteilverfahren ermöglichen es, eine geheime Nachricht so aufzuteilen, dass diese an Hand einer ausreichenden Anzahl an Teilen korrekt rekonstruiert werden kann, nichts jedoch über dessen Inhalt in Erfahrung zu bringen ist, falls zu wenige Teile vorhanden sind. Homomorphe Verfahren erlauben es zudem, auf den Teilen Berechnungen durchzuführen um dadurch Teile der Summen und Produkte von Geheimnissen zu erhalten – und lassen dadurch eine Vielzahl an Anwendungen zu, darunter auch die sogenannte sichere Mehrparteienberechnung. Bei dieser können unterschiedliche Parteien gemeinsam eine beliebige Funktion berechnen, ohne dass sie etwas über die Eingabewerte der jeweils Anderen erfahren.

Um die Größe der erzeugten Teile in einem Geheimnisteilverfahren zu reduzieren, wurden sogenannte „computational"-sichere Varianten entwickelt, die den Schutz vor einer illegitimen Aufdeckung des Geheimnisses nur gegen Angreifer garantieren, deren Berechnungszeit durch ein Polynom in der Länge der Eingabe begrenzt ist. Während diese Verfahren in Bezug auf den Speicherplatzbedarf deutlich effizienter sind, haben sie jedoch den Nachteil, nicht die genannten homomorphen Eigenschaften aufzuweisen. In dieser Arbeit wird auf der Basis vollhomomorpher Verschlüsselung ein Geheimnisteilverfahren entwickelt, dass beide Vorteile vereint und somit speicherplatzeffizient ist, aber dennoch die Möglichkeit einräumt auf den Teilen der Geheimnisse rechnen zu können. Dafür benötigt das Verschlüsselungsverfahren die Homomorphie-Eigenschaft auch für Chiffrate, die mit unterschiedlichen Schlüsseln erzeugt wurden. Aus diesem Grund schließt die Arbeit ein Nachweis dieser Eigenschaft für eine verbesserte Variante eines solchen Verschlüsselungsverfahrens ein.

Der zweite Teil der Arbeit führt zunächst in die Theorie der algebraischen Funktionenkörper und den darauf basierenden algebraisch-geometrischen Codes ein. Für diese ist der Riemann-Roch'sche Satz zentral, der im Zuge einer Einführung erläutert wird. Grundlage des Teils bildet eine Veröffentlichung von Cascudo, Cramer und Xing [CCX12b], in der eine unendliche Familie von homomorphen, bzw. „d-arithmetischen" (d. h. mit d-fachen Multiplikationen kompatiblen) Geheimnisteilverfahren mit asymptotisch guten Eigenschaften konstruiert wird. Die Konstruktion des zugrundeliegenden Codes erfolgt dabei an Hand unendlicher Klassenkörpertürme unter Ausnutzung eines gruppentheoretischen Arguments von Golod und Šafarevič. Damit dieser für den Zweck der Geheimnisteilung geeignet ist, werden dessen definierende Divisoren auf der Grundlage eines sogenannten Riemann-Roch-Gleichungssystems gewählt. Diese Systeme bilden eine Verallgemeinerung bisheriger Verfahren und ermöglichen es, die d-Torsion der Grad-Null-Divisorenklassengruppe einzubeziehen.

Die so konstruierten d-arithmetischen Geheimnisteilverfahren werden zudem als Baustein für die Konstruktion des oben genannten computational-sicheren d-arithmetischen Verfahrens eingesetzt. Im Nachgang wird beschrieben, wie dieses Verfahren, dessen Linearität von der in der Literatur üblichen Definition abweicht, für die sichere Mehrparteienberechnung eingesetzt werden kann. Darüber hinaus wurde darauf geachtet, den Zusammenhang zwischen den an den Anforderungen beweisbarer Sicherheit orientierten Begriffen und der algebraisch formulierten Codierungstheorie herauszustellen.

Contents

Introduction

Keeping a secret secure involves two tasks at the same time. First, you have to make sure that no-one can get hold of the secret and second you have to store it in a way that it cannot get lost. The obvious solutions to both problems seem to contradict each other, that is hiding a secret would make it small and inaccessible which is in opposition to making it redundant and storing it in different places. This becomes clear in a corporate setting where giving an essential cryptographic key to only one person might lead to a total loss of data once the person becomes unavailable, while giving the key to larger number of employees will make it more likely to leak.

Shamir [S79] and Blakley [B79] came up with an elegant solution to this problem: a secret sharing scheme (SSS). In such a scheme, so-called shares are generated from the secret and distributed to different players. However, each player individually cannot learn the secret from his share, but a (pre-specified) set of shares are needed for the reconstruction of the secret. In an (n, t)-threshold scheme, any t out of n shares can reconstruct the secret, but with less, nothing about the secret can be learned. Thereby we get both, redundancy as loss of up to $n - t$ shares can be coped with, and secrecy, as at least t shares have to leak to get hold of the secret.

In short, Shamir's scheme works by randomly drawing the coefficients of a polynomial of degree t over $\mathbb{F}_q$ ($q > n$ a prime-power) and setting the secret as its constant term. The n shares are generated by evaluating the polynomial at publicly known (non-zero) places and distributed by the dealer of the scheme. The reconstruction works as an application of Lagrange's interpolation theorem, as the polynomial is completely determined by a set of $t + 1$ shares, i. e., points of the polynomial, but the value at the place zero can still be chosen freely if no more than t shares are given. A more complete description is given in Section 1.1.

Shamir's secret sharing scheme has been improved and generalized in several ways and in this process it has become clear that quite a number of applications can be based on secret sharing. Beimel [B11] gives an excellent survey on the topic and names secure multiparty computation (MPC), Byzantine agreement, attribute-based encryption, and generalized oblivious transfer as prominent applications. For more general access conditions, we can formulate the reconstructability in terms of qualified coalitions of the players. Such a collection $\mathcal{A} \subseteq \mathcal{P}(P)$ of *qualified* subsets of the player set P specifies which coalitions of participants are able to recover the secret from their shares. Moreover, a second collection $\mathcal{B} \subseteq \mathcal{P}(P)$ of *unqualified* subsets specifies the coalitions not able to reconstruct the secret, due to the secrecy condition of the scheme. We call $(\mathcal{A}, \mathcal{B})$ an *access structure* and note that they are monotone, because if a player set in $\mathcal{A}$ can access the secret, then so does any superset, and analogously if a player set in $\mathcal{B}$ cannot learn anything about the secret, then so does any of its subsets. Therefore, it suffices to specify the min terms of $\mathcal{A}$ and the max terms of $\mathcal{B}$. A scheme and its access structure are called *perfect*, if $\mathcal{B} = \overline{\mathcal{A}}$. For a well-written survey on the combinatorial aspects of access structures see [P12].

Ito, Saito, and Nishizeki [ISN89] gave the first construction of these general, perfect schemes, but the size of the shares given to each player may be exponential in the number of participants. Later, Benaloh and Leichter [BL90] improved this with a recursive construction on a (monotone) formula defining $\mathcal{A}$. Any access structure can be given as such a formula, but as the number of monotone formulas is doubly exponential in the number of participants, only those secret sharing schemes given by a small formula are efficient. Here, we call a scheme *efficient*, if the size of its shares is polynomial in the number of participants.

Because secret sharing schemes are mostly used in homomorphic settings such as MPC, nearly all schemes devised so far are *linear secret sharing schemes* (LSSS) over a finite field $\mathbb{F}_q$, where the *reconstruction* and *distribution functions* are linear over $\mathbb{F}_q$. These schemes have been shown to be equivalent in [KW93; B96] to so-called *monotone span programs* (MSP), which can be seen as a labeled matrix representation. Note that these can also be defined over general rings instead of finite fields, leading to a similar correspondence of secret sharing schemes over a ring R and MSPs over the integers [C+03; CF02]. Here, only black-box access to addition, multiplication and random sampling methods of R are required.

One issue in the theory of secret sharing is its space efficiency or information rate. While there are several ways to weaken the security of the scheme to obtain an improved space efficiency, not all are practical for usual applications of secret sharing. For instance, so-called *computational secret sharing* (CSS) schemes—where an unqualified player set can learn something about the secret, if they would be able to break the security of an encryption scheme—are space-optimal but have the downside of not being sufficiently homomorphic, which makes them impractical for the use in higher-order protocols like MPC. We aim to overcome this limitation by using the recently developed fully homomorphic encryption (FHE) schemes.

Speaking of efficiency, a small MSP leads to an efficient LSSS. Moreover, there are some access structures for which the MSP representation is much smaller than the formula representation. This gives constructions of efficient schemes for an enlarged number of access structures.

The space efficiency of a secret sharing scheme is given by its information ratio, which can be defined as the ratio of the length of the longest share and the length of the secret. While it is conjectured [B11] that there are perfect access structures which admit schemes only with an information ratio exponential in n, the best currently known lower bound is $\Omega(n/\log n)$, as shown by Csirmaz [C97]. However, due to its equivalence to MSP it has been shown by Gàl [G01] that for the special case of LSSS there is a superpolynomial $2^{O(\log n)}$ lower bound. Note that in perfect secret sharing schemes, any share is as least as large as the secret. If all shares are exactly the size of the secret, that is, the information ratio is 1, the scheme is called *ideal*. These schemes have been characterized in terms of matroids, see [P12] for reference.

When the total amount of data exchanged is at least n times the secret size, the scheme becomes impractical for large secrets. One way to overcome this limitation is to consider non-perfect secret sharing schemes, where no secrecy or reconstruction guarantees are made for coalitions neither in $\mathcal{A}$, nor in $\mathcal{B}$. The most extreme case is an *information dispersal algorithm* (IDA), which is a secret sharing scheme with $\mathcal{B} = \varnothing$, first defined by Rabin [R89]. For $\mathcal{A} = \mathcal{A}_t$, where $\mathcal{A}_t$ is the collection of sets with cardinality t, it holds that the information ratio is $1/t$, which is the theoretical optimum. Another approach to overcome the problem of efficiency is to lower the general secrecy requirements. While in usual schemes information-theoretic privacy is assumed, we devise schemes which only guarantee secrecy against computationally bounded attackers. These *computational secret sharing* (CSS) schemes protect against polynomially bounded probabilistic attackers and were first defined by Krawczyk [K94] for the threshold case. Béguin and Cresti [BC95] later generalized the construction to arbitrary perfect access structures. The efficiency of computationally secure schemes resemble those of information dispersal algorithms, plus a small term which depends only on the security parameter. Therefore, CSS schemes are much more efficient than their information-theoretic counterparts. For a formulation of the corresponding security notions, Bellare and Rogaway [BR07] present a modern provable-security approach to CSS which also aims to improve the original scheme of [K94].

Moreover, in most practical settings, other cryptographic primitives such as encryption or pseudorandom functions are used in the surrounding protocol, for which security is based on the same weaker assumptions. Therefore, it makes perfect sense to prefer computational secret sharing over schemes which are secure in the information-theoretic sense in these settings. The construction given in [K94; BC95] is a simple combination of encryption and information dispersal: First, the secret is encrypted with a secret key, which is shared by a perfect information-theoretically secure scheme, while the encrypted secret is distributed through the size-optimal IDA. While we devise this variant in a homomorphic way, we include also an analysis of the variant which first shares the secret and encrypts it afterwards.

Secret sharing schemes can be seen as a simple primitive to construct *verifiable secret sharing* (VSS) schemes and *secure multiparty computation* protocols (MPC), i. e., protocols where n players jointly compute a previously agreed function on their inputs and during the process have a guarantee of the privacy of their inputs and correctness of the result if not too many players try to cheat. Cramer, Damgård, and Maurer [CDM00] showed that any perfect LSSS can be transformed into a multiplicative one (which allows running a multiplication protocol) while only doubling its size. Moreover, from any perfect LSSS for which the access structure fulfills certain conditions, a generic construction for VSS and MPC protocols is given. In [C$^+$03] this is extended to secret sharing schemes over arbitrary rings. An alternative version of such a construction of MPC is given by [M03], which uses less preconditions on the side of the secret sharing scheme. As the schemes devised in this thesis satisfy a slightly weaker version of linearity (namely its reconstruction map, but not necessarily its share map is linear) and the results of [CDM00; C$^+$03] only hold for LSSS with a linear share map, we point out a construction of these protocols in our setting.

For our construction of homomorphic CSS schemes, we use fully homomorphic encryption schemes (FHE), i. e., encryption schemes which admit any arithmetic circuit to be executed on the input ciphertexts by a special Eval-function of the scheme. After decryption, the resulting plaintext is the outcome of the circuit executed on the input plaintexts. López-Alt, Tromer, and Vaikuntanathan [LTV12] showed that this is possible even for plaintexts encrypted under different keys and devised a so-called multikey FHE based on NTRU. An improved, but non-multikey version of their scheme was recently given by Bos et al. [B^+13].

For the second part of the thesis, we note that LSSS over a finite field can also be based on algebraic geometric codes. For example, Chen and Cramer [CC06] constructed LSSS over $\mathbb{F}_q$ based on these codes. Their scheme is strongly multiplicative and can be used for MPC, without the restriction of $n > q$ as in Shamir's scheme. Subsequent work [C^+08; C^+09; CCX11] improved on these results. They are analyzed in Chapter 3, and we refer to the beginning of the chapter for a more thorough introduction.

Contribution

- Creation of the first *computational* secret sharing scheme with homomorphic properties, such as strong multiplicativity and arithmetic secret sharing. (Chapter 2)
- The scheme is suitable for passively secure multiparty computation. Moreover, we point out a method to turn it into the full adaptively secure version using replicated secret sharing. (Section 2.2)
- Review of the literature on secret sharing and algebraic geometric codes with focus on the question of Riemann–Roch equation systems for the construction of asymptotically good secret sharing schemes. (Chapter 3)
- Description of a slightly generalized version of linear and arithmetic secret sharing schemes in terms of a provable-security framework. (Section 1.2)
- Deduction of refined noise bounds for the multi-key variant of a fully homomorphic encryption scheme. (Section 1.6.1)

Related Work

Kikuchi et al. [K^+13] try to combine the share efficiency of a CSS scheme with the homomorphism properties of LSSS by using a conversion protocol between the two worlds. Their scope is different as they do not devise a homomorphic CSS scheme but instead use a non-linear CSS for storing the secret and convert it to a LSSS in the event that a calculation is to be performed on the shares. For this, they use the share conversion method of Cramer, Damgård, and Ishai [CDI05].

As homomorphic properties are difficult to achieve for CSS schemes, an analysis of computational VSS schemes by Backes, Kate, and Patra [BKP11] showed that they could as well be based on non-homomorphic commitments.

Furthermore, Tate, Vishwanathan, and Weeks [TVW13] look into the question of encrypting shares consistently with a public key encryption scheme and introduce the method of plaintext randomization to make their formal proofs rigorous.

Moreover, note that MPC can be based directly on fully homomorphic encryption, as in [LTV12], who devise a scheme which minimizes interaction during the computation to the starting and reconstruction phase of the protocol, using a multikey FHE scheme. In this context, we also refer to [A$^+$12] for an MPC protocol based on FHE.

Outline

In Chapter 1 we review some preliminaries of secret sharing with a focus on homomorphic properties and computational secret sharing. Furthermore we present the relevant part of the literature on fully homomorphic encryption and their multikey variants. In Chapter 2 we then give our construction of the first homomorphic CSS scheme and point out how it can be used in VSS and MPC protocols. In Chapter 3 we introduce the reader to basic results on algebraic function fields including the Riemann–Roch theorem and review the relevant literature of secret sharing schemes based on algebraic geometry codes. We take special care on an construction aspect of the codes used in [CCX11]. In the last chapter we conclude our research and look into the question of future work.

1 Preliminaries

In this chapter, we introduce basic notions of secret sharing in a provable security framework, including linearity and multiplicativity, in a slightly generalized version, which is fitted for our construction in the later part of the thesis. Furthermore, we present central notions of secure multiparty computation, including the universal composability framework of Canetti [C01] as reviewed in [CD05] in Sections 1.3 to 1.5. Finally, homomorphic encryption is introduced in Section 1.6, as it is a key component for our construction in Chapter 2.

For notation, we write vectors in bold, $\boldsymbol{s}_T$ denotes the vector restricted to index set $T \subseteq I$. We denote the length of a string m as $|m|$ and write 1^κ for the string of length κ consisting of only the letter 1. If χ is a probability distribution, we write $x \leftarrow \chi$ to indicate that x is sampled from χ. Analogously by $x \leftarrow S$ we state that x is sampled according to the uniform distribution on the set S. In the same fashion, we write $x \leftarrow A(\cdot)$, if x is the outcome of a possibly randomized algorithm A. Furthermore, we denote with $\mathrm{negl}(\kappa)$ a function which is *negligible* in κ, i. e., it is less than $1/p(\kappa)$ for any polynomial p and sufficiently large κ. We assume that $\mathbb{N}$ includes zero. Note that whenever we speak of a ring, we assume that it is commutative and with 1.

1.1 Secret Sharing Schemes

In this section we introduce secret sharing schemes and their properties. In our presentation we mostly follow the definitions of Bellare and Rogaway [BR07], who suggested a powerful framework of modern provable-security secret sharing incorporating a number of notions defined in the field. We recommend [B11] as a comprehensive survey on secret sharing schemes with information-theoretic privacy.

Let $P = \{P_1, \dots, P_n\}$ be a set of players. An n-player *distribution scheme* with message space $\mathcal{M}$ and share spaces $\mathcal{S}_1, \dots, \mathcal{S}_n$ is a pair $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$, where

1. $\mathsf{Sh}\colon \mathcal{M} \to \prod_{i=1}^{n} \mathcal{S}_i$ is a probabilistic algorithm returning an n-vector $\boldsymbol{s}$ on input $m \in \mathcal{M}$,

2. $\mathsf{Rec}\colon \prod_{i=1}^{n}(\mathcal{S}_i \cup \{\Diamond\}) \times \{0, \dots, n\} \to \mathcal{M} \cup \{\perp\}$ is a deterministic algorithm returning a value m on input $(\boldsymbol{s}, j)$, where $j \in \{0, \dots, n\}$ specifies whether an honest participant P_j, $j \in \{1, \dots, n\}$ does the reconstruction, or whether external reconstruction is performed, if $j = 0$. The entries of the vector $\boldsymbol{s}$ may contain a distinguished sign $\Diamond$ to indicate that the entry is omitted in the reconstruction process. If Rec is unable to reconstruct an $m \in \mathcal{M}$, it returns $\perp$.

A so-called *dealer* uses the distribution scheme Σ to generate a *share vector* $\boldsymbol{s}$ from a message $m \in \mathcal{M}$ and then sends entry $\boldsymbol{s}_i$ to player P_i. If the players wish to reconstruct the secret afterwards, they gather enough shares needed for reconstruction and then a player P_j runs the reconstruction procedure $\mathsf{Rec}(\boldsymbol{s}, j)$ to retain the original message. Lost shares or shares omitted in the gathering process are denoted by $\Diamond$.

The second argument in the Rec algorithm is given to model the difference between internal and external reconstruction: when reconstruction is done by a participant, he has more information than an external party, as he knows that he himself is honest. However, this plays a role in robust reconstruction guarantees only, cf. Definition 1.2.

A distribution scheme with certain privacy and reconstruction guarantees is called a secret sharing scheme (SSS). Before we turn to a formal definition, we are in need of a way to express these guarantees in mathematical terms. Therefore, let P be the set of players and denote its powerset by $\mathcal{P}(P)$. We define the *access structure* of a secret sharing scheme on P as a pair $(\mathcal{A}, \mathcal{B})$, where $\mathcal{A} \subseteq \mathcal{P}(P)$ contains the *qualified* and $\mathcal{B} \subseteq \mathcal{P}(P)$ the *unqualified* player coalitions and $\mathcal{A} \cap \mathcal{B} = \varnothing$. That is, any collection of players $X \in \mathcal{P}(P)$ can either jointly reconstruct a shared secret, if $X \in \mathcal{A}$, or it should not be possible for them to learn anything about the secret (in a sense specified later), if $X \in \mathcal{B}$. Note that in order to guarantee that the distributed message can be reconstructed at all, we require that $P \in \mathcal{A}$.

A scheme and its access structure are called *perfect*, if $\mathcal{B} = \overline{\mathcal{A}}$. In non-perfect schemes, it is possible that X is neither in $\mathcal{A}$ nor in $\mathcal{B}$; in this case nothing can be said in advance about the ability to learn or reconstruct the secret. Note that for any access structure $\Gamma = (\mathcal{A}, \mathcal{B})$, $\mathcal{A}$ is monotonically increasing and $\mathcal{B}$ is monotonically decreasing, due to the nature of secret sharing. Therefore, it suffices to specify the minimal qualified sets of $\mathcal{A}$ and the maximal unqualified sets of $\mathcal{B}$, denoted as $\min \mathcal{A}$ and $\max \mathcal{B}$, respectively. An important special case of these structures are *threshold access structures*, where any set with $r \geq 1$ or more elements can reconstruct the message, while no set of less than or equal to t elements is qualified. We denote it by $\Gamma(t, r)$.

While access structures are essential and classical in the definition of secret sharing properties, we follow [BR07] in looking at the topic from a modern provable-security standpoint. Therefore, we use adversaries represented by Turing machines (TMs) which try to break the guarantees of the secret sharing in a game-based approach, to specify the exact security conditions. An adversary is called *probabilistic polynomial time* (PPT), if it may use randomness and is polynomially time-bounded in the input length. During the game, the adversary may use a corruption oracle which allows it to obtain the share of the specified players.

With respect to the access structure of a secret sharing scheme, we can now define an adversary as a TM which tries to violate certain guarantees of the scheme in a game-like setting. So we define a *$\mathcal{B}$-privacy adversary A* to be an adversary as in Experiment 1.1, who tries to distinguish two self-chosen equal-length messages by only looking at the shares of a set of players that it is allowed to corrupt, i. e., of a set in $\mathcal{B}$. Its advantage in the game of Experiment 1.1, relative to the security parameter κ, is denoted as $\mathbf{Adv}^{\text{priv}}_{\Sigma,A}(\kappa)$.

Analogously, we define an *$\mathcal{A}$-reconstruction adversary A* to be an adversary as in Experiment 1.2, trying to prohibit the correct reconstruction of the message, by removing shares or injecting new share values for players it may corrupt. This is specified by

$\mathcal{A}$, namely, A must leave at least one set in $\mathcal{A}$ uncorrupted. Moreover, to model the reconstruction as done by a honest player, who can be sure that his share is correct, A may output additionally a player index j. For the adversary definition, we require that if $j \neq 0$, the share of player P_j is unchanged. We define the advantage of A in the game of Experiment 1.2 relative to the security parameter κ as $\mathbf{Adv}^{\mathrm{rec}}_{\Sigma,A}(\kappa)$.

For some adversary models, this is too powerful. For a weaker version of a $\mathcal{B}$-reconstruction adversary, consider the restriction that A may only learn and erase the shares of a player upon corruption. In formal terms, for such an adversary it holds that $\boldsymbol{s}'_{i \in T}$ of Experiment 1.2 is in $\{\boldsymbol{s}_i, \Diamond\}$. In this case, we call A an *$\mathcal{B}$-reconstruction erasure adversary*—in opposition to *substituting adversaries*, who may inject an arbitrary share to corrupted players.

For the formal definitions of the adversaries' advantage, we set

$$\mathbf{Adv}^{\mathrm{priv}}_{\Sigma,A}(\kappa) := 2 \cdot \Pr\left[\mathbf{Exp}^{\mathrm{priv}}_{\Sigma,A}(\kappa) = 1\right] - 1, \text{ and} \tag{1.1}$$

$$\mathbf{Adv}^{\mathrm{rec}}_{\Sigma,B}(\kappa) := 2 \cdot \Pr\left[\mathbf{Exp}^{\mathrm{rec}}_{\Sigma,B}(\kappa) = 1\right] - 1, \tag{1.2}$$

where $\mathbf{Exp}^{\mathrm{priv}}_{\Sigma,A}(\kappa)$ and $\mathbf{Exp}^{\mathrm{rec}}_{\Sigma,B}(\kappa)$ are defined in Experiment 1.1 and Experiment 1.2, respectively. In both cases, a call of the corruption oracle $\mathsf{corrupt}(\boldsymbol{s}, i)$ adds player P_i to the set T of corrupted players and returns P_i's share of the secret.

Definition 1.1. An n-player *secret sharing scheme* is a distribution scheme $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ with access structure $\Gamma = (\mathcal{A}, \mathcal{B})$, satisfying the following conditions:

Privacy. For any $\mathcal{B}$-privacy adversary A it holds

- $\mathbf{Adv}^{\mathrm{priv}}_{\Sigma,A}(\kappa) = 0$ for *perfect (information-theoretic)* privacy.
- $\mathbf{Adv}^{\mathrm{priv}}_{\Sigma,A}(\kappa) = \mathrm{negl}(\kappa)$ for *statistical* privacy.
- $\mathbf{Adv}^{\mathrm{priv}}_{\Sigma,A}(\kappa) = \mathrm{negl}(\kappa)$ if A is PPT, for *computational* privacy.

Reconstructability. For any $\mathcal{A}$-reconstruction erasure adversary A it holds

- $\mathbf{Adv}^{\mathrm{rec}}_{\Sigma,A}(\kappa) = 0$.

Note that we distinguish two types of perfectness. As specified above, $\Gamma = (\mathcal{A}, \mathcal{B})$ is called perfect, if $\mathcal{A} = \overline{\mathcal{B}}$. However, we also call the privacy of Σ perfect, if $\mathbf{Adv}^{\mathrm{priv}}_{\Sigma,A}(\kappa) = 0$. In the first case, we call Σ perfect, in the second case, we call Σ a perfect-privacy SSS. Moreover, to guarantee reconstruction in the presence of substitution adversaries, we also give the corresponding definitions for robust secret sharing.

Definition 1.2 (Robust secret sharing)**.** Let Σ be an n-player secret sharing scheme as in Definition 1.1. Σ is called *robust* if additionally the following holds.

Robust reconstructability. For any $\mathcal{A}$-reconstruction (substituting) adversary A it holds

- $\mathbf{Adv}^{\mathrm{rec}}_{\Sigma,A}(\kappa) = 0$ for *perfect* robustness.
- $\mathbf{Adv}^{\mathrm{rec}}_{\Sigma,A}(\kappa) = \mathrm{negl}(\kappa)$ for *statistical* robustness.

```
$(m_0, m_1, state) \leftarrow A_1(1^\kappa)$
$b \leftarrow \{0,1\}$
$\boldsymbol{s} \leftarrow \mathsf{Sh}(m_b)$
$b^* \leftarrow A_2^{\mathsf{corrupt}(\boldsymbol{s},\cdot)}(1^\kappa, state)$
if $b = b^*$ then
    return 1
else
    return 0
```

Experiment 1.1. The priv experiment of secret sharing scheme Σ with adversary $A = (A_1, A_2)$ challenged to compromise the privacy of Σ by distinguishing the sharing of one of two equal-length, self-chosen secrets, corrupting only players of a set in $\mathcal{B}$. To allow a PPT adversary, which is polynomially bounded in its input length, a number of steps polynomial in the security parameter κ, we pass it a unary coding of κ. The convention to write the oracle map in the exponent of the adversary is used to indicate its availability towards the adversary.

- $\mathbf{Adv}^{\text{rec}}_{\Sigma,A}(\kappa) = \text{negl}(\kappa)$ if A is PPT, for *computational* robustness.

Although robust secret sharing has stronger reconstructability guarantees, this notion does not provide security against a dishonest dealer. For example, the dealer might distribute shares that cannot be reconstructed to any valid secret, or he might implement a way to corrupt shares for the reconstruction to reveal a different secret. As secret sharing is often used as part of a larger protocol, e. g., in establishing secure multiparty computation (MPC) against active adversaries, each player may assume the dealer role in different steps of the protocol. To overcome the problems with dishonest dealers in MPC protocols, the dealer has to commit to having shared a certain value, resulting in verifiable secret sharing (VSS) schemes, as described in Section 1.4.

1.2 Linear and Multiplicative Secret Sharing Schemes

Most of the currently devised secret sharing schemes are linear secret sharing schemes over a finite field or a ring, due to its suitability for applications, such as secure multiparty computation. In such schemes, linear operations on the shares are compatible with the sharing and are preserved upon reconstruction. We formally define it as follows.

Let $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ be an n-player secret sharing scheme with message space $\mathcal{M}$ and share spaces $\mathcal{S}_1, \dots, \mathcal{S}_n$. We set $\mathcal{S} \coloneqq \prod_i \mathcal{S}_i$. Formally, we call Σ a *linear secret sharing scheme* (LSSS) for $\mathcal{M}$ over Λ, if Λ is an associative R-algebra, i. e., R is a ring and there is a ring homomorphism $\varphi\colon R \to \Lambda$, $\mathcal{M}$ is a finitely-generated Λ-module, $\mathcal{S}_i$ $(i = 1, \dots, n)$ are finitely-generated R-modules and for all $a \in R$, $\boldsymbol{s}$, $\boldsymbol{s'} \in \mathcal{S}$ in the image of Sh, and $j \in \{0, \dots, n\}$, we have

$$\mathsf{Rec}(a \cdot \boldsymbol{s} + \boldsymbol{s'}, j) = \varphi(a) \cdot \mathsf{Rec}(\boldsymbol{s}, j) + \mathsf{Rec}(\boldsymbol{s'}, j).$$

```
$(m, state) \leftarrow A_1(1^\kappa)$
$\boldsymbol{s} \leftarrow \mathsf{Sh}(m)$
$(\boldsymbol{s}', j) \leftarrow A_2^{\mathsf{corrupt}(\boldsymbol{s},\cdot)}(1^\kappa, m, state)$
if $\mathsf{Rec}(\boldsymbol{s}_{\overline{T}} \sqcup \boldsymbol{s}'_T, j) \neq m$ then
    return 1
else
    return 0
```

Experiment 1.2. The rec experiment of secret sharing scheme Σ with adversary $A = (A_1, A_2)$ aiming at preventing the correct reconstruction of a self-chosen secret, leaving at least a set of players in $\mathcal{A}$ uncorrupted. Note that T is the set of corrupted players, and j is an uncorrupted player the adversary may point out explicitly, depending on the security notion. If $j = 0$, an external reconstruction has to be performed, leading to stronger guarantees; see [BR07] for details. Note that $\boldsymbol{s}_{\overline{T}} \sqcup \boldsymbol{s}'_T$ is the vector $\boldsymbol{x}$ with $\boldsymbol{x}_i = \boldsymbol{s}_i$ if $i \notin T$ and $\boldsymbol{x}_i = \boldsymbol{s}'_i$ if $i \in T$.

Written in more algebraic terms, we can express this as the property that

$$\mathsf{Rec}(\cdot, j)|_{\mathsf{im}\,\mathsf{Sh}} \colon \mathcal{S} \to \varphi^*(\mathcal{M})$$

is an R-linear map for any $j \in \{0, \dots, n\}$, where, $\varphi^*(\mathcal{M})$ denotes the *restriction of scalars* of $\mathcal{M}$, i. e., $\mathcal{M}$ with the same additive group, but with R-scalar multiplication $R \times \mathcal{M} \to \mathcal{M}$, $(a, s) \mapsto \varphi(a)s$. Here, we restrict Rec to the image of Sh to avoid an algebraic treatment of the empty share sign $\Diamond$, although it is still assumed linear when projected to the components which do not contain $\Diamond$, provided that reconstruction is possible.

As Sh is a probabilistic algorithm, we can also make the used randomness explicit by looking at the corresponding deterministic algorithm $\mathsf{Sh} \colon \mathcal{M} \times \mathcal{R} \to \mathcal{S}$, where $r \in \mathcal{R}$ is sampled according to the uniform random distribution on the set $\mathcal{R}$. Note that we will assume that $\mathcal{R}$ is a finitely-generated and free Λ-module.

Remark 1.1. In the case usually found in the literature, we have that $R = \Lambda$, $\varphi = \mathrm{id}_R$ and $\mathsf{Rec}(\cdot, j)|_{\mathsf{im}\,\mathsf{Sh}}$ is an R-linear map of free and finitely-generated R-modules. This implies that there is an equivalent LSSS of the same size, such that also $\mathsf{Sh} \colon \mathcal{M} \times \mathcal{R} \to \mathcal{S}$ is an R-linear map. The so-obtained Sh is a *section* for Rec, i. e., a linear map such that $\mathsf{Rec}(\cdot, j) \circ \mathsf{Sh} = \mathrm{id}_{\mathcal{M}}$, which always exists for free (or projective) modules. For the case of R a field, compare also [B96]. These linear schemes over rings or fields are equivalent to so-called monotone span programs defined in [KW93], as shown in [B96] and [CF02], respectively.

However, it is important to note that this is no longer the case in our more general setting. This can be seen in the case where $\mathcal{M} = \Lambda$ and $\mathcal{S} = R$, as the R-module homomorphism $\varphi \colon \mathcal{S} \to \mathcal{M}$ need not have a section. To see this, let $\mathcal{M} = \mathbb{Z}/2\mathbb{Z}$ and $\mathcal{S} = \mathbb{Z}/4\mathbb{Z}$. Any $s \colon \mathcal{M} \to \mathcal{S}$ would have to send $1 \mapsto 2$ as 2 is the only element of order two in $\mathcal{S}$, but $\varphi(2) = \varphi(1) + \varphi(1) = 0$, hence $\varphi \circ s = 0 \neq \mathrm{id}_{\mathcal{M}}$.

Remark 1.2. Let $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ be an n-player SSS with message space $\mathcal{M}$ and access structure $\Gamma = (\mathcal{A}, \mathcal{B})$. Note that because of $P \in \mathcal{A}$, the reconstruction property of the SSS implies that $\mathsf{Rec}(\cdot, j) \circ \mathsf{Sh} = \mathrm{id}_{\mathcal{M}}$, for $j \in \{0, \dots, n\}$. Moreover, if $\mathcal{M}$ and $\mathcal{S}_i$ fulfill the same conditions as in the definition of LSSS, and Sh is a linear map, then the reconstruction map $\mathsf{Rec}(\cdot, j)\big|_{\operatorname{im}\mathsf{Sh}} \colon \mathcal{S} \to \mathcal{M}$ is also linear, for any $j \in \{0, \dots, n\}$.

We have the following classical example of an LSSS, namely the scheme of Shamir [S79], modified with an additional condition on the ω_i to also work for rings, cf. [C$^+$03, Proposition 1].

Example 1.1 (Shamir's scheme for rings)**.** Let R be a ring without zero divisors and $R^\times$ denote its invertible elements. We assume there exist $\omega_1, \dots, \omega_n \in R^\times$ with $\omega_i - \omega_j \in R^\times$ for all $i \neq j$. Then there is a LSSS over R for $\Gamma = \Gamma(t, n-t)$, which works as follows.

In order to share a secret $m \in R$ the dealer chooses $a_1, \dots, a_{t-1} \leftarrow R$ uniformly at random and defines a polynomial $p \in R[T]$ of degree $t-1$ as $p := m + \sum_{i=1}^{t-1} a_i T^i$. The shares are then obtained by evaluating p at the publicly known places $\omega_1, \dots, \omega_n \in R^\times$, i. e., the share of player P_i is set as $p(\omega_i)$, for $i = 1, \dots, n$.

Reconstruction works by the Lagrange interpolation theorem, as t points on p suffice to reconstruct it uniquely, and then evaluate $p(0) = m$. For this, assume that $\boldsymbol{s}_i := p(\omega_i)$, and that we have entries $\boldsymbol{s}_{i_1}, \dots, \boldsymbol{s}_{i_t}$. We compute the Lagrange polynomial form $q \in R[T]$ of p as

$$q = \sum_{j=1}^{t} \boldsymbol{s}_{i_j} \prod_{j \neq k} \frac{\omega_{i_k} - T}{\omega_{i_k} - \omega_{i_j}},$$

using only the given places and the publicly known ω_i. As it agrees with p at $\omega_{i_1}, \dots, \omega_{i_t}$, and is of degree $t-1$, it is identical to p. Moreover, by assumption, the expression in the denominator is not equal to zero. So, evaluating $q(0) = m$ we obtain the secret message.

Moreover, q is linear in the shares and the privacy condition follows as for $t-1$ points, we can artificially add another one with coordinates $(0, m')$, to obtain a uniquely determined polynomial $\tilde{q}$ with $\tilde{q}(0) = m'$. To obtain a valid share for this, we then have to evaluate $\tilde{q}(\omega_{i_t})$, for an index i_t whose entry was not previously known.

1.2.1 Multiplicative Secret Sharing

This section is about LSSS that exhibit multiplicativity and we therefore assume additionally that the message space $\mathcal{M}$ is an Λ-algebra, i. e., additionally to being an Λ-module, it has a Λ-bilinear multiplication operation. Furthermore, the share spaces $\mathcal{S}_1, \dots, \mathcal{S}_n$ are R-algebras with R-bilinear multiplications $\circledast_i$.

Definition 1.3. Let $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ be an n-player LSSS over Λ with message space $\mathcal{M}$, share spaces $\mathcal{S}_i$ as described above, and access structure $\Gamma = (\mathcal{A}, \mathcal{B})$. Σ is said to be *multiplicative*, if for all $j \in \{0, \dots, n\}$, m, $m' \in \mathcal{M}$ with $\boldsymbol{s} \leftarrow \mathsf{Sh}(m)$, $\boldsymbol{s}' \leftarrow \mathsf{Sh}(m')$, it holds:

$$m \cdot m' = \mathsf{Rec}(\boldsymbol{s} * \boldsymbol{s}', j), \text{ where } \boldsymbol{s} * \boldsymbol{s}' := \sum_i \mathsf{Sh}'_i(\boldsymbol{s}_i \circledast_i \boldsymbol{s}'_i),$$

where $\mathsf{Sh}'_i\colon \mathcal{S}_i \to \mathcal{S}$ is a share map working directly on the share space of player P_i. (We can set $\mathsf{Sh}'_i \coloneqq \mathsf{Sh} \circ \varphi_i$, where $\varphi_i\colon \mathcal{S}_i \to \mathcal{M}$ is a lift of the ring homomorphism $\varphi\colon R \to \Lambda$ to the corresponding modules; assume that $\mathcal{S}_i$ and $\mathcal{M}$ are free over R and Λ for this purpose.) Moreover, Σ is said to be *strongly multiplicative* if the condition holds, even if $\boldsymbol{s} * \boldsymbol{s'}$ is restricted to a qualified player set $A \in \mathcal{A}$ (containing player P_j, if $j \neq 0$), i. e., the usual reconstruction property of the secret sharing scheme holds also for these products.

The significance of this definition is as follows: We aim to define a protocol that allows us to obtain a situation, given that two messages are already shared, that each player holds a share of the product of the two messages, without intermediate reconstruction. For this note that $\boldsymbol{s_i} \circledast_i \boldsymbol{s'_i}$ can be calculated by a local computation, i. e., player P_i can compute it using only his own shares of the two messages. After this, each player P_i, $i = 1, \dots, n$ can share their locally computed products, leading to new share vectors $\boldsymbol{t_1}, \dots, \boldsymbol{t_n}$. This step is called the *resharing step*. By the linearity of the scheme, we can add these locally again, to obtain a share of the sum, which is, by definition of the multiplicativity of the scheme, a share of the product.

Let us remark a few things on the special share map Sh'_i. In the case usually found in the literature we have, as mentioned before, $\Lambda = R$ and $\varphi = \mathrm{id}_R$. In the case of $\mathcal{M} = R = \mathcal{S}_1 = \ldots = \mathcal{S}_n$, we further note that $\varphi_i = \varphi = \mathrm{id}_R$, so that $\mathsf{Sh}'_i = \mathsf{Sh}$. In this case the locally computed product of the shares lies directly again in the message space, so there are no hurdles for the resharing step. In our more general situation this is not the case, so the natural solution is to map it to the message space using the ring homomorphism φ associated to our scheme. However, in our computational secret sharing scheme, to be constructed in Chapter 2, φ is a decryption function, which can only be executed with knowledge of the secret key, which we do not have in this situation. This issue will be solved in the corresponding section by using a different but compatible share map which works in R, see Section 2.1.1.

Note that strongly multiplicative schemes have the following restrictions on the access structure of a scheme. Let Γ be an access structure on player set P. For this, we define the element-wise union of set systems as: $\mathcal{A}_1 \sqcup \mathcal{A}_2 \coloneqq \{A_1 \cup A_2\colon A_1 \in \mathcal{A}_1, A_2 \in \mathcal{A}_2\}$. We state the generalized version of the Q^3 property as in [FM02]: Γ is said to be Q^3, if $P \notin \overline{\mathcal{A}} \sqcup \mathcal{B} \sqcup \mathcal{B}$. This is necessary for strong multiplication and we assume it tacitly for the rest of the thesis, whenever we have the strong multiplication property.

Example 1.2. The LSSS of Example 1.1 is multiplicative, if and only if $t < n/2$, and strongly multiplicative, if and only if $t < n/3$, cf. [C+03, Proposition 1].

Remark 1.3 (d-fold products)**.** We can generalize the definition of strong multiplicativity to d-fold products, in a straightforward manner. For this let $j \in \{0, \dots, n\}$, $m_1, \dots, m_d \in \mathcal{M}$ with $\boldsymbol{s_i} \leftarrow \mathsf{Sh}(m_i)$, $i = 1 \dots, d$ be arbitrary. Then it has to hold

$$m_1 \cdots m_d = \mathsf{Rec}(\boldsymbol{s_1} * \cdots * \boldsymbol{s_d}, j), \text{ where } \boldsymbol{s_1} * \cdots * \boldsymbol{s_d} \coloneqq \sum_i \mathsf{Sh}'_i(\boldsymbol{s_1}_i \circledast_i \cdots \circledast_i \boldsymbol{s_d}_i),$$

even when $\boldsymbol{s_1} * \cdots * \boldsymbol{s_d}$ is restricted to a set of $\mathcal{A}$ (containing player P_j, if $j \neq 0$). While we can already compute any circuits with strong multiplicativity alone, this stronger property avoids resharings steps in between, so that we can compute d-fold products with only one resharing step afterwards.

1.2.2 Games for Homomorphic Secret Sharing Schemes

For linear secret sharing scheme without a linear Sh function, we have the problem that privacy and reconstruction guarantees do not need to hold in general for sums and products of shares. In order to get a fine-grained control on the guarantees after arithmetic operations, we introduce classes of circuits $\mathcal{C}$ and the corresponding modified games for privacy and reconstruction adversaries, parametrized by a circuit class as $\mathcal{C}$-priv and $\mathcal{C}$-rec. Moreover, we describe a general method to compatibly lift a circuit from a Λ-algebra $\mathcal{M}$ to the share space of the secret sharing scheme. At the end of the section we state a certain property of the scheme, which allows reduce the $\mathcal{C}$-privacy and $\mathcal{C}$-reconstruction guarantees to the "ordinary" privacy and reconstruction guarantees.

Definition 1.4 (Algebraic Circuits)**.** Let $t \in \mathbb{N}_{\geq 1}$. We define a t-ary *algebraic circuit over a ring* Λ as an directed acyclic graph with the following properties: a) there are t nodes with in-degree zero, called input nodes, b) there is one node with out-degree zero, called output node, c) any other node is called a *gate* and is labeled by one of the ring operations. We might restrict ourself to the case where each gate has in-degree two.

The size of the circuit is the number of its gates, and its depth is the length of the longest path from any input node to the output node. A circuit C as described naturally gives rise to a map $C(\cdot)\colon \Lambda^t \to \Lambda$. Analogously, we can define circuits with l output nodes and circuits for modules or algebras over a ring, where each input node is labeled, stating whether it accepts inputs from the ring of scalars or from the module or algebra.

We propose the following naming conventions for classes of circuits on an algebra over a ring. Denote by $\mathcal{C}_{t \to l}$ the class of all t-ary circuits with addition, scalar multiplication and multiplication gates and l output nodes. Moreover, denote the $\mathcal{C}^{\mathrm{lin}}$ the class of linear circuits, i. e., without multiplication gates. Moreover, $\mathcal{C}^{\leq L}$ denotes the class of all circuits of size polynomial in the number of input nodes, and maximal depth L. Furthermore, we set $\mathcal{C}^{\mathrm{lin}}_{t \to l} \coloneqq \mathcal{C}_{t \to l} \cap \mathcal{C}^{\mathrm{lin}}$ and $\mathcal{C}^{\leq L}_{t \to l} \coloneqq \mathcal{C}_{t \to l} \cap \mathcal{C}^{\leq L}$.

To lift a circuit specified on our message space $\mathcal{M}$, which is an algebra over the ring Λ, to the share space $\mathcal{S}$, the addition and scalar multiplication gates are directly mapped to addition and scalar multiplication operation gates on the share space. However, for multiplication to work, we need a resharing step, as described in the previous section on multiplicative secret sharing. For this we map a multiplication gate to a combination of n gates using the local multiplication $\circledast_i$ on $\mathcal{S}_i$, to which we append a structure of addition and scalar multiplication gates representing the share function in the resharing process. Afterwards, we have an multi fan-in addition gate (or its representation as multiple gates with fan-in 2), which take as input all the outputs of the share structures. Note that the share "block" needs randomness as additional input to work correctly.

If Σ is compatible with d-fold products, then we need only one resharing step after any d multiplications. We can simply assume that $\mathsf{Lift}(C)$ chooses the maximal possible d for this, to avoid unnecessary resharing steps, for usual reconstruction guarantees. If we would like to have a more robust reconstruction, we can decrease the number of multiplications before a resharing step, accordingly.

We denote the map we described in the previous paragraph by $\mathsf{Lift}(C, r)$, which lifts a circuit C defined on $\mathcal{M}$ to $\mathcal{S}$, and takes additional input $r \in \mathcal{R}$ for the resharing steps. We write $\mathsf{Lift}(C)$ for short, if we want to leave the randomness implicit, as usual.

Definition 1.5. We call an adversary A a *$\mathcal{C}$-arithmetic $\mathcal{B}$-privacy adversary*, if it is defined as in Experiment 1.3 and tries to distinguish the sharing of two evaluations of a self-chosen t-ary circuit $C \in \mathcal{C}$ on the share vectors of equal-length, self-chosen secrets by only looking at the shares of a set of players that it is allowed to corrupt, i. e., of a set in $\mathcal{B}$. Its advantage in the game of Experiment 1.3, relative to the security parameter κ, is denoted as $\mathbf{Adv}^{\mathcal{C}\text{-priv}}_{\Sigma,A}(\kappa)$. Moreover, we define a *$\mathcal{C}$-arithmetic $\mathcal{A}$-reconstruction adversary* A to be an adversary as in Experiment 1.4, trying to prohibit the correct reconstruction of the evaluation of a self-chosen t-ary circuit $C \in \mathcal{C}$ on self-chosen secrets, by removing shares or injecting new share values for players it may corrupt. Here again, A must leave at least one set in $\mathcal{A}$ uncorrupted and if A outputs a player index $j \neq 0$, the share of player P_j is unchanged. We define the advantage of A in the game of Experiment 1.2 relative to the security parameter κ as $\mathbf{Adv}^{\mathcal{C}\text{-rec}}_{\Sigma,A}(\kappa)$. $\mathcal{C}$-arithmetic $\mathcal{A}$-reconstruction erasure adversaries are defined analogously as in Section 1.1.

Remark 1.4 (concerning evaluation keys)**.** Note that, as it is the case in our secret sharing scheme of Chapter 2, the evaluation of a circuit on the share space may require so-called *evaluation keys*. If this is the case, we modify the secret sharing scheme, so that it saves the evaluation keys corresponding to the shares, and exhibits an additional map $\mathsf{EvalKey}$ that outputs the evaluation key corresponding to a share vector. For arbitrary share vectors to work, it might require a reference to the original share vectors as output by the scheme and a description of the calculation used to obtain it, in form of a circuit.

We can describe the statement that the whole player set P can reconstruct the secret messages after evaluation of circuits $C \in \mathcal{C}$, as the property of the following diagram to be commutative:

$$\begin{tikzcd}
\mathcal{M}^t \arrow[r, "\bigtimes_{i=1}^{t} \mathsf{Sh}"] \arrow[d, "C"] & \mathcal{S}^t \arrow[d, "\mathsf{Lift}(C)"] \\
\mathcal{M}^l & \mathcal{S}^l \arrow[l, "\bigtimes_{i=1}^{l} \mathsf{Rec}"]
\end{tikzcd}$$

Definition 1.6. Let Σ be a linear secret sharing scheme over Λ with message space $\mathcal{M}$, and $\mathcal{C}$ a non-empty class of circuits on $\mathcal{M}$. Σ is said to be *$\mathcal{C}$-arithmetic*, if the following holds:

1. If there is a circuit in $\mathcal{C}$ containing a multiplication gate, then there is an R-algebra structure on the share spaces,

```
$(m_0^1, \dots, m_0^t, m_1^1, \dots, m_1^t, C, state) \leftarrow A_1(1^\kappa)$
$b \leftarrow \{0, 1\}$
$\boldsymbol{s_i} \leftarrow \mathsf{Sh}(m_b^i)$, for $i = 1, \dots, t$
$ek_i = \mathsf{EvalKey}(\boldsymbol{s_i})$, for $i = 1, \dots, t$
$\boldsymbol{s} \leftarrow \mathsf{Lift}(C)((\boldsymbol{s_1}, ek_1), \dots, (\boldsymbol{s_t}, ek_t))$
$b^* \leftarrow A_2^{\mathsf{corrupt}(\boldsymbol{s}, \cdot)}(1^\kappa, ek_1, \dots, ek_t, state)$
if $b = b^*$ then
    return $1$
else
    return $0$
```

Experiment 1.3. The $\mathcal{C}$-priv experiment of secret sharing scheme Σ with adversary $A = (A_1, A_2)$ challenged to compromise the privacy of Σ by distinguishing the sharing of two evaluations of a self-chosen t-ary circuit $C \in \mathcal{C}$ on the share vectors of equal-length (i. e., $|m_0^i| = |m_1^i|$, $i = 1, \dots, t$), self-chosen secrets, corrupting only players of a set in $\mathcal{B}$. Here, $\mathcal{C}$ is a distinguished class of circuits on $\mathcal{M}$ and ek_i are evaluation keys, generated by Σ, as they are needed in order to evaluate C, cf. Remark 1.4.

2. For any $\mathcal{C}$-arithmetic $\mathcal{B}$-privacy adversary A of Σ, there is a $\mathcal{B}$-privacy adversary B of Σ, such that B is PPT if A is, and
$$\mathbf{Adv}_{\Sigma,A}^{\mathcal{C}\text{-priv}}(\kappa) \leq \mathbf{Adv}_{\Sigma,B}^{\text{priv}}(\kappa).$$

3. For any $\mathcal{C}$-arithmetic $\mathcal{B}$-reconstruction erasure adversary A of Σ, there is a $\mathcal{B}$-reconstruction erasure adversary B of Σ, such that B is PPT if A is, and
$$\mathbf{Adv}_{\Sigma,A}^{\mathcal{C}\text{-rec}}(\kappa) \leq \mathbf{Adv}_{\Sigma,B}^{\text{rec}}(\kappa).$$

If Σ is robust and item 3 holds analogously for substituting adversaries, then Σ is said to have *robust $\mathcal{C}$-arithmetic reconstruction.*

In our search for criteria which allow us to show $\mathcal{C}$-arithmeticity of a secret sharing scheme, we define the following property.

Definition 1.7 (circuit privacy)**.** Let $\mathcal{C}$ be a circuit class over $\mathcal{M}$ and $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ an n-player LSSS over Λ with message space $\mathcal{M}$ and access structure $\Gamma = (\mathcal{A}, \mathcal{B})$, with an R-algebra structure on the share spaces, if there is a circuit in $\mathcal{C}$ containing a multiplication gate. Let $\mathbf{Exp}_{\Sigma,A}^{\text{wc-priv}}(\kappa)$ be the game which an adversary A wins, if it is successful in distinguishing the evaluation of two self-chosen t-ary circuits of $\mathcal{C}$ with the same number of output nodes on self-chosen messages $m_1, \dots, m_t \in \mathcal{M}$ when given access to the shares of players of a set in $\mathcal{A}$ and optionally the corresponding evaluation keys.

In other words, it distinguishes the distributions $\{\mathsf{Lift}(C_1)((m_1, ek_1), \dots, (m_t, ek_t))\}$ and $\{\mathsf{Lift}(C_2)((m_1, ek_1), \dots, (m_t, ek_t))\}$ restricted to a set in $\mathcal{A}$ that A may adaptively determine, by corrupting players as needed. Denote the advantage of A in the game by

```
$(m_1, \dots, m_t, C, state) \leftarrow A_1(1^\kappa)$
$\boldsymbol{s_i} \leftarrow \mathsf{Sh}(m_i)$, for $i = 1, \dots, t$
$ek_i = \mathsf{EvalKey}(\boldsymbol{s_i})$, for $i = 1, \dots, t$
$\boldsymbol{s} \leftarrow \mathsf{Lift}(C)((\boldsymbol{s_1}, ek_1), \dots, (\boldsymbol{s_l}, ek_t))$
$(\boldsymbol{s'}, j) \leftarrow A_2^{\mathsf{corrupt}(\boldsymbol{s},\cdot)}(1^\kappa, m_1, \dots, m_t, C, ek_1, \dots, ek_t, state)$
if $\mathsf{Rec}(\boldsymbol{s}_{\overline{T}} \sqcup \boldsymbol{s}'_T, j) \neq C(m_1, \dots, m_t)$ then
    return 1
else
    return 0
```

Experiment 1.4. The $\mathcal{C}$-rec experiment of secret sharing scheme Σ with adversary $A = (A_1, A_2)$ aiming at preventing the correct reconstruction of the evaluation of a self-chosen t-ary circuit $C \in \mathcal{C}$ on self-chosen secrets, leaving at least a set of players in $\mathcal{A}$ uncorrupted. Note that T is the set of corrupted players, and j is an uncorrupted player the adversary may point out explicitly, depending on the security notion. Here, $\mathcal{C}$ is a distinguished class of circuits on $\mathcal{M}$ and ek_i are evaluation keys, generated by Σ, as they are needed in order to evaluate C, cf. Remark 1.4.

$\mathbf{Adv}^{\text{wc-priv}}_{\Sigma,A}(\kappa)$. Then Σ is said to have *weak perfect circuit privacy*, if $\mathbf{Adv}^{\text{wc-priv}}_{\Sigma,A}(\kappa) = 0$ for any such adversary A and weak *statistical* circuit privacy, if $\mathbf{Adv}^{\text{wc-priv}}_{\Sigma,A}(\kappa) = \text{negl}(\kappa)$. Moreover, the *computational* circuit privacy guarantee restricts the adversary class additionally to PPT.

Analogously, we define $\mathbf{Exp}^{\text{sc-priv}}_{\Sigma,A}(\kappa)$ to be the game as before, but instead of distinguishing the evaluations of two different circuits, A has to distinguish the evaluation of a self-chosen circuit C on the share space, from an evaluation of C on the message space, with the sharing applied afterwards, i. e., it needs to distinguish $\{\mathsf{Sh}(C(m_1, \dots, m_t))\}$ and $\{\mathsf{Lift}(C)((m_1, ek_1), \dots, (m_t, ek_t))\}$. We call this the strong circuit privacy game and define the corresponding *strong* circuit privacy notions, as expected.

Lemma 1.1. *Let $\mathcal{C}$ be a circuit class for $\mathcal{M}$ and $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ an n-player LSSS over Λ with message space $\mathcal{M}$, with an R-algebra structure on the share space $\mathcal{S}$, if there is a circuit in $\mathcal{C}$ containing a multiplication gate. If Σ has the strong circuit privacy property for $\mathcal{C}$ as described above, then it holds that for any $\mathcal{C}$-arithmetic $\mathcal{B}$-privacy adversary A of Σ, there is a $\mathcal{B}$-privacy adversary B of Σ, such that B is PPT if A is, and $\mathbf{Adv}^{\mathcal{C}\text{-priv}}_{\Sigma,A}(\kappa) \leq \mathbf{Adv}^{\text{priv}}_{\Sigma,B}(\kappa)$.*

Proof. This is a straightforward reduction. For this, let B make use of a simulation of adversary A as follows: When A outputs $m_0^1, \dots, m_0^t$, $m_1^1, \dots, m_1^t$ and a t-ary circuit $C \in \mathcal{C}$, it simply calculates $m_0 := C(m_0^1, \dots, m_0^t)$ and $m_1 := C(m_1^1, \dots, m_1^t)$ and sends these out as part of the priv game. Whenever A makes use of its corruption oracle, B queries its own for the same player, and forwards the share entry to A. The final output of A is then returned by B to the surrounding game. Due to strong circuit privacy A cannot tell whether it obtains shares for which the sharing procedure was applied after the evaluation of C or whether it was later performed on the share spaces, as in the usual $\mathcal{C}$-privacy game, and so cannot make use of the fact. □

It would be interesting to extend the previous lemma to weak circuit privacy. We leave this as an open problem for now and turn to a corollary for schemes with a linear share map.

Corollary 1.1. *Let $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ be an n-player LSSS over Λ with message space $\mathcal{M}$ with linear Sh map. Then Σ is $\mathcal{C}^{\text{lin}}$-arithmetic. If Σ is robust then it also has robust $\mathcal{C}^{\text{lin}}$-arithmetic reconstruction.*

Proof. The $\mathcal{C}^{\text{lin}}$-privacy property is immediate by Lemma 1.1, as a linear share map leads to strong circuit privacy for $\mathcal{C}^{\text{lin}}$. Moreover the $\mathcal{C}^{\text{lin}}$-reconstruction guarantee follows directly from the definition of an LSSS. □

Lemma 1.2. *Let $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ be a strongly multiplicative n-player LSSS over Λ with message space $\mathcal{M}$. Let $\mathcal{C} := \mathcal{C}_{t \to l}$. Then it holds that for any $\mathcal{C}$-arithmetic $\mathcal{B}$-reconstruction erasure adversary A of Σ, there is a $\mathcal{B}$-reconstruction erasure adversary B of Σ, such that B is PPT if A is, and $\mathbf{Adv}^{\mathcal{C}\text{-rec}}_{\Sigma,A}(\kappa) \leq \mathbf{Adv}^{\text{rec}}_{\Sigma,B}(\kappa)$.*

Proof. This is immediate by induction as the invariant of reconstructability is preserved after a multiplication with resharing by definition. □

1.3 Universal Composability

Our aim is to use secret sharing schemes to realize general secure multiparty computation protocols, which do not leak private information and assure correctness of the computation, regardless of the context it is used in. Of course, to be meaningful, we have to take real-world adversarial powers in account, which is typically the corruption of participants. When the adversary corrupts a player, he learns all of its previous in- and output, and, if he is *active*, can control all future actions. A *passive adversary* is restricted to learning the information and cannot alter the behavior of players.

Players and adversaries are modeled by Turing machines and therefore the security has to be proven against all possible attacks. In the past, it was not so rare that attempts to capture the adversarial behavior overlooked harmful actions or protocol states induced by its role in a higher-level protocol.

However, in this section we introduce an approach to overcome these problems, namely the universal composability (UC) framework, based on the ideal- vs real-world approach. In our definitions and presentation of the protocols, we follow [C01; N03] as reviewed in Cramer and Damgård [CD05]. For the basic setting we assume a synchronous network between players $P_1, \ldots, P_n$ and the adversarial environment Z. These are modeled as interactive Turing machines (ITMs), which have additional tapes used for communication, and are connected by pairwise secure channels (in the information theoretic setting). Moreover, a secure broadcast channel for each player is assumed.

Note that the environment Z is defined as the adversary, having full control over everything that is not directly part of the protocol as run by the players. The aim is to design our real-world protocol in a way, that Z is unable to distinguish a run of our protocol from an ideal-world setting, which is per definition free of unwanted information leakage, etc. For example, the ideal world can contain an incorruptible third party, to which players send their inputs, and which never leaks anything of it.

More precisely, after running either the protocol, or the ideal functionality, whose interaction is presented to Z by a PPT simulator S to look like a run of the real-world protocol, the environment outputs a bit, guessing which of the two worlds it is in. If it is unable to distinguish the two, we can regard the real protocol as secure as the incorruptible ideal-world functionality, leading to proof of security against any adversarial action and independent of the context the real-world protocol is used in. Therefore we can later use this ideal functionality as a blackbox in a higher-level protocol, without compromising our security guarantees.

Definition 1.8. Let F be an ideal functionality, and π a real-world protocol. We say that π *securely realizes* F, if, there is a PPT simulator S, such that for every adversarial environment Z and every input z,

$$\mathbf{Adv}^{\mathrm{UC}}_{\pi,F,S,Z}(\kappa, z) := \Pr\Big[\mathbf{Exp}^{\mathrm{real}}_{\pi,Z}(\kappa, z) = 1\Big] - \Pr\Big[\mathbf{Exp}^{\mathrm{ideal}}_{F,S,Z}(\kappa, z) = 1\Big],$$

is negligible in κ, where $\mathbf{Exp}^{\mathrm{real}}_{\pi,Z}(\kappa, z)$ and $\mathbf{Exp}^{\mathrm{ideal}}_{F,S,Z}(\kappa, z)$ are just a run of the protocol, with a guess of Z whether it is in the ideal or the real-world-scenario, afterwards. We call the security computational, if Z is additionally assumed to be a PPT.

1.4 Linear Distributed Commitments

In this section we describe the ideal functionality F_{HC} which stores homomorphic commitments and releases them again on an opening request of the committing player, in the synchronous setting. This functionality is essential for establishing verifiable secret sharing and multiparty computation protocols secure against active attacks. Note that our presentation of the relevant commands follows Cramer and Damgård [CD05].

A *commitment* protocol allows a player to fix a certain value a without revealing it to anybody else (the so-called hiding property). Later, he can execute an opening command, showing the previously fixed value to the other participants. For this, he is bound to a and cannot influence the protocol to reveal any other value except a or a fail signal (the binding property).

We denote a commitment of value a by player P_i as $[a]_i$, which consists of a correct (or rather consistent, that is recoverable) sharing of a. The commitment scheme is called *linear*, if from u, $[a]_i$ and $[b]_i$ a commitment of $[a + b]_i$ and $[u \cdot a]_i$ can be generated without any interaction. It is called a *distributed* commitment scheme, as it works via a secret sharing process, in contrast to cryptographic commitments which use certain encryption schemes. Hence, it is necessary for the real-world protocol to have all honest players participating. This fact is modeled by the requirement that a command is issued by all honest players cooperatively in the ideal world.

Note that any implementation will guarantee security only when the protocol is used as intended. So, for example, a disagreement of honest players will cause F_{HC} to broadcast all internal data and stop working. In this case it becomes trivial to simulate and we do not have to care about the unintended use in our proof of security for a given real-world protocol. Note that we use the *synchronous* model for our setting.

We specify the commands *commit*, *open*, *add*, *cmult* below, together with three additional commands for *commitment transfer* (CTP), *commitment sharing* (CSP) and a check whether a third commitment contains the product of two others (CMP). The round numbers in the command specifications below are all relative to the current round of the command initiation. Moreover, the identifier variables id_x are for unique identification of the committed values in the database of the ideal functionality. To ensure correct use of identifiers, we would suggest a consecutive numbering, which clarifies for all involved parties which id_x should be used.

Commit. Here, we describe the protocol interface of player P_i committing a value a. Let $d_c \in \mathbb{N}$ denote the commitment delay, which gives a corrupted player the possibility to change his committed value after the initiation of the command.

1. In the round 1, every player P_j, $i \neq j$ sends a message (commit, i, id), while P_i sends $(\text{commit}, i, id, a)$.

2. In round $2, \dots, d_c - 1$, a corrupt P_i may once send $(\text{commit}, i, id, a')$ to change the commitment to value a'.

3. In round d_c, the functionality sends $(\text{commit}, i, \text{success})$ to every player, unless a or $a' = \perp$. In this case, $(\text{commit}, i, \text{fail})$ is send. Afterwards, either (i, id, a) or (i, id, a') is stored, depending on whether P_i made use of the possibility to override the committed value.

Open. This is the opening protocol broadcasting a of the previously stored commitment $[a]_i$. Note that a private opening to only a player P_j is possible, when the opening command is accommodated with an additional parameter j.

1. In round 1, all players send (open, i, id) to F_{HC}, while P_i may send an additional "refuse" or "accept".

2. In round 2, F_{HC} sends a message (open, id, a) containing the stored value a of the commitment to every player, unless P_i sent "refuse". In this case $(\text{open}, id, \text{fail})$ is sent.

Addition. All honest players send $(\text{add}, id_1, id_2, id_3)$ in the same round. If (i, id_1, a), (i, id_2, b) have been stored previously, then F_{HC} stores $(i, id_3, a + b)$. This computes $[a + b]_i$ from $[a]_i$ and $[b]_i$ without interaction, as it can be typically done in a real-world protocol by a local computation on the players side.

Constant Multiplication. All honest players send $(\text{cmult}, id_1, id_2, u)$ in the same round. If (i, id_1, a), has been stored previously, then F_{HC} stores $(i, id_2, u \cdot a)$. This computes $[u \cdot a]_i$ from $[a]_i$ and a factor u without interaction.

Commitment Transfer. This command transfers a commitment from player P_i to P_j, i. e., creating a new commitment $[a]_j$ from a previously stored $[a]_i$. Let d_{ctp} denote the CTP delay which gives a corrupted commitment owner the possibility to send a refuse signal, causing a fail of the commitment transfer.

1. In the round 1, all honest players send $(\text{ctp}, i, id_1, j, id_2)$.
2. In round $2, \ldots, d_{ctp} - 1$, a corrupt P_i may send (id_1, refuse).
3. In round d_{ctp}, F_{HC} sends $(id_1, id_2, \text{fail})$ to every participant, if P_i opted to refuse and $(id_1, id_2, \text{success})$ otherwise. In the later case, it stores (j, id_2, a) and sends the value a privately to P_j.

Commitment Sharing and VSS. The commitment sharing protocol performs a secret sharing on a committed value a, i. e., creates shares $v_1, \ldots, v_n$ based on the valid secret sharing parameters or randomness (e. g., polynomial coefficients chosen by P_i) and stores commitments $[v_1]_1, \ldots, [v_n]_n$. Let d_{csp} denote the CSP delay which gives a corrupted P_i time to change the secret sharing parameters after the initiation of the command.

1. In round 1, all honest players send $(\text{csp}, id_0, \ldots, id_n)$ to F_{HC}. If (i, id_0, a) is the corresponding stored commitment and P_i is honest, he also sends the parameters needed to secret share a.
2. In round $2, \ldots, d_{csp} - 1$, a corrupt P_i may change the secret sharing parameters, or send a message $(id_0, \ldots, id_n, \text{refuse})$.
3. In round d_{csp}, either send $(id_0, \ldots, id_n, \text{fail})$ (when a refuse signal was received) or store (j, id_j, v_j), where v_j is the share of player P_j, as shared by the received parameters. In the same round, broadcast $(id_0, \ldots, id_n, \text{success})$. In this process, player P_j learns the committed share v_j.

Note that verifiable secret sharing (VSS) is just the process of first committing to a value, and then sharing it via the CSP protocol. This guarantees that the sharing is correct, each player is committed to their share (hence, a robust sharing) and the shares can be reconstructed, resulting in the value the dealer has committed herself to.

Commitment Product Test. The commitment multiplication protocol (CMP) allows to test whether a commitment $[c]_i$ contains the product of two other specified commitments $[a]_i$ and $[b]_i$, i. e., whether $c = a \cdot b$. This is needed to assure that corrupted players cannot change the result of an interactive multiplication, without failing on this check procedure afterwards. Here, d_{cmp} denotes the delay allowing a corrupt owner of the commitments to send a refuse signal.

1. In round 1, all honest players send $(\text{cmp}, id_1, id_2, id_3)$.
2. In round $2, \ldots, d_{cmp} - 1$, a corrupt P_i may send $(id_1, id_2, id_3, \text{refuse})$.

3. In round d_{cmp}, if (i, id_1, a), (i, id_2, b) and (i, id_3, c) have been previously stored, P_i did not send a refuse command before, and $a \cdot b = c$, F_{HC} sends a confirmation message $(id_1, id_2, id_3, \text{success})$ to everyone. Otherwise, $(id_1, id_2, id_3, \text{fail})$ is send.

Remark 1.5 (Protocol Instances based on LSSS)**.** Note that, as shown in [CD05; C+03], generic protocols for CMP, CTP, and CSP exist, which are based on the addition, constant multiplication, commit and open commands. See also [FM02]. However, they make use of the property, that the Sh-function of the scheme is linear, in contrast to our setting. We will discuss an adaption to our setting in Section 2.2.

1.5 Secure Multiparty Computation

In the following we describe the ideal functionality F_{MPC} [CD05, p. 51] for secure multiparty computation in the synchronous communication setting. Assume that $\boldsymbol{x}_i = \perp$, for $i = 1, \ldots, n$. Let d_i denote the input delay (number of rounds the adversary can change input values) and d_c denote the computation delay (number of rounds the computation takes place).

1. In round 1, collect input messages. If all honest players sent their input, set $\boldsymbol{x}_i = v$ for all P_i, and send an "Inputs received" message on the corrupt output port. Otherwise, abort. (Note that, when aborting, all internal data is sent to the corrupt output port.)

2. In rounds $2, \ldots, d_i$, corrupt players may send a change message, which resets x_i to a new value v'. On reception of any message from a honest player after round 1, abort.

3. After $d_i + d_c$ rounds, set $(y_1, \ldots, y_n) = f(x_1, \ldots, x_n)$ and send y_i to P_i.

This setting with an input delay $d_i \geq 1$ allows the adversary to be *rushing*, i. e., it may decide its input, after the honest players fixed their input and can afterwards use the information of corrupted players, to decide on further corruptions and value changes of already corrupted players, provided these still fulfill the conditions of the adversary structure.

In accordance to Definition 1.8, we can then say, that a real-world protocol π *securely realizes* F_{MPC}, if there is a PPT simulator S, such that for every adversarial environment Z and every input z,

$$\mathbf{Adv}^{\mathrm{UC}}_{\pi, F_{\mathrm{MPC}}, S, Z}(\kappa, z) := \Pr\Big[\mathbf{Exp}^{\mathrm{real}}_{\pi, Z}(\kappa, z) = 1\Big] - \Pr\Big[\mathbf{Exp}^{\mathrm{ideal}}_{F_{\mathrm{MPC}}, S, Z}(\kappa, z) = 1\Big],$$

is negligible in κ, where $\mathbf{Exp}^{\mathrm{real}}_{\pi, Z}(\kappa, z)$ and $\mathbf{Exp}^{\mathrm{ideal}}_{F_{\mathrm{MPC}}, S, Z}(\kappa, z)$ are just a run of the protocol, with a guess of Z whether it is in the ideal or the real-world-scenario, afterwards.

A passively secure MPC-protocol based on LSSS. Using the notation we introduced above, we can simply describe a passively secure MPC protocol π based on an n-player linear secret sharing scheme Σ which is $\mathcal{C}$-arithmetic for a circuit class $\mathcal{C}$. For this, in the first round all players share their inputs and distribute their shares to the other players. As no active corruptions occur, we can set $d_i = 1$. Let $C \in \mathcal{C}$ be the n-ary circuit to be evaluated, which we want to execute on the share vectors of the inputs, to preserve their privacy. For this, we proceeding exactly like the $\mathsf{Lift}(C)$ map, setting the share vectors of the players as the values of the input nodes of the lifted circuit. Any addition and constant multiplication gates can be computed locally, and for the multiplication gates we have to do the resharing steps as prescribed in $\mathsf{Lift}(C)$. In the end, the final share vector is reconstructed and yields the result.

Theorem 1.1. *The protocol π realizes F_{MPC} in the information-theoretic scenario with computational security against an adaptive and passive environment corrupting at most a set in $\mathcal{B}$, and with $d_i = 1$, d_c equal to the depth of the circuit used to implement the function computed.*

Proof. See [CD05, Theorem 2]. □

1.6 Homomorphic Encryption

Homomorphic encryption schemes allow to perform operations on ciphertexts. Upon decryption, we retain a message which is the result of the corresponding operations in plaintext space. Moreover, the ciphertexts do not grow too much in this process.

While such a scheme exhibits a plethora of applications, including for example, cloud computing with privacy guarantees, and was therefore much looked for, the first such scheme (based on an approximation problem over lattices) was devised in 2009 in the groundbreaking work of Gentry [G09]. Until then, researchers came up with a second generation of schemes with a significant improvement on performance and practicality, see e. g., [BV11a; BV11b; FV12; BGV12]. Moreover, López-Alt, Tromer, and Vaikuntanathan [LTV12] introduced the notion of a multikey homomorphic encryption scheme, allowing operations on ciphertexts encrypted under different keys. Before we describe a variant of their scheme in Section 1.6.1, we discuss general notions of homomorphic encryption schemes in the following.

Signature of Multikey Homomorphic Encryption. Let $N \in \mathbb{N}_{\geq 1}$ and $\mathcal{C}$ be a class of circuits on a ring Λ. An N-key $\mathcal{C}$-homomorphic encryption scheme $\mathsf{HE} := \mathsf{HE}^{(N)} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ with message space Λ and ciphertext space R has the following signature:

- $\mathsf{HE.Keygen}(1^\kappa)$: Generates the public parameters depending on the security parameter κ and from them, it creates a key pair consisting of the public key pk, and the secret key sk, accompanied with an evaluation key ek, which is needed to evaluate homomorphic operations on ciphertexts. It outputs (pk, sk, ek).

- $\mathsf{HE.Enc}(pk, m)$: Encrypts the message $m \in \Lambda$ under public key pk.

- $\mathsf{HE.Dec}(sk_1, \ldots, sk_k, c)$: Decrypts a ciphertext $c \in R$ using a set of secret keys. If c is the result of an evaluation on ciphertexts encrypted under public keys corresponding to $sk_1, \ldots, sk_k$, all these secret keys are needed to retain the plaintext.

- $\mathsf{HE.Eval}(C, (c_1, ek_1), \ldots, (c_k, ek_k))$: Evaluates a k-ary circuit $C \in \mathcal{C}$ on ciphertexts $c_1, \ldots, c_k$. For this, the corresponding evaluation keys are needed and a ciphertext $c^* \in R$ is returned. (In contrast to [LTV12], we do not require that the pk_i are passed to Eval. This is done to support secret-key variants more easily.)

Correctness. Let $\mathsf{HE} := \mathsf{HE}^{(N)}$ be as above. For the scheme to be correct, it has to hold for all $C \in \mathcal{C}$, all k-length subsequences $\{(pk_i, sk_i, ek_i)\}_{i \in \{1,\ldots,k\}}$ of all sequences of tuples $\{(pk'_i, sk'_i, ek'_i)\}_{i \in \{1,\ldots,N\}}$ of length N in the support of $\mathsf{HE.Keygen}(1^\kappa)$, all plaintexts $m_1, \ldots, m_k \in \Lambda$ and all ciphertexts $c_1, \ldots, c_k \in R$, such that c_i is in the support of $\mathsf{HE.Enc}(pk_i, m_i)$, it holds that if

$$c^* = \mathsf{HE.Eval}(C, (c_1, ek_1), \ldots, (c_k, ek_k)),$$

then

$$\mathsf{HE.Dec}(sk'_1, \ldots, sk'_N, c^*) = C(m_1, \ldots, m_t).$$

In other words, after evaluating a circuit $C \in \mathcal{C}$ on ciphertexts, the decryption of the resulting c^* should be possible, when given the corresponding secret keys, and should yield the result of C applied directly to the plaintexts. This holds, even if given additional keys not used in the encryption.

By a hybrid argument, correctness can be shown to also hold for circuits with l output nodes. Written in a different way, this implies that the following diagram commutes for any circuit $C \colon \Lambda^k \to \Lambda^l \in \mathcal{C}$, k, $l \in \mathbb{N}_{\geq 1}$:

$$\begin{tikzcd}
\Lambda^k \arrow[r, "\bigtimes_{i=1}^{k} \mathsf{HE.Enc}(pk_i, \cdot)"] \arrow[d, "C"] & R^k \arrow[d, "{\mathsf{HE.Eval}(C, (\cdot, ek_1), \ldots, (\cdot, ek_k))}"] \\
\Lambda^l & R^l \arrow[l, "{\mathsf{HE.Dec}(sk_1, \ldots, sk_k, \cdot)^l}"]
\end{tikzcd}$$

Remark 1.6. If $\mathcal{C}$ is the class of all circuits on Λ, C_{add} and C_{mult} are the addition and multiplication circuits, respectively, sk_1, sk_2 and ek_1, ek_2 are secret and evaluation keys associated to the ciphertexts, then $\mathsf{HE.Dec}(sk_1, sk_2, \cdot) \colon R \to \Lambda$ is a ring homomorphism compatible with the structures $(R, \oplus, \odot)$ and $(\Lambda, +, \cdot)$, where $\oplus$ is given by $\mathsf{HE.Eval}(C_{\mathrm{add}}, (\cdot, ek_1), (\cdot, ek_2))$ and $\odot$ by $\mathsf{HE.Eval}(C_{\mathrm{mult}}, (\cdot, ek_1), (\cdot, ek_2)))$. Here C_{add} and C_{mult} denotes the circuit with a single addition and multiplication gate, respectively. Note that this *does not* (generally) hold for other structures on R.

Compactness. Let HE, C, $\{(pk_i, sk_i, ek_i)\}_{i \in \{1,\ldots,k\}}$, $m_1, \ldots, m_k \in \Lambda$, and $c_1, \ldots, c_k$ be as in the preceding paragraph. For the scheme to be *compact*, it has to hold that if $c^* = \mathsf{HE.Eval}(C, (c_1, ek_1), \ldots, (c_k, ek_k))$, there is a polynomial p, such that

$|c^*| \leq p(\kappa, N)$. This implies that the length of the resulting ciphertext does not depend on k or C and avoids trivial solutions where just $(C, c_1, \ldots, c_k)$ is returned on evaluation.

Note that it is open, whether it is possible to avoid the dependence of p on N. Compactness is a crucial property for fully homomorphic encryption and together with correctness, it leads to the following definition.

Definition 1.9 ([LTV12])**.** A family $\{\mathsf{HE}^{(N)} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})\}_N$ of encryption schemes as above, with $N \in \mathbb{N}_{\geq 1}$, is *multikey $\mathcal{C}$-homomorphic*, if it is both correct and compact for all $N \in \mathbb{N}_{\geq 1}$ and $C \in \mathcal{C}$. Any $\mathsf{HE}^{(N)}$ is called a N-key $\mathcal{C}$-homomorphic encryption scheme. We speak of a *fully homomorphic* scheme or family, if $\mathcal{C}$ is the class of all circuits.

Security. We state the common notion of passive security of homomorphic encryption schemes, when given an additional evaluation key ek which is necessary for evaluation of circuits. Note that we cannot hope for the stronger IND-CCA2 (chosen-ciphertext attack) security, due to the general incompatibility of non-malleability and homomorphic encryption.

Definition 1.10 (IND-CPA security)**.** Let $\mathsf{HE} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be an N-key $\mathcal{C}$-homomorphic encryption scheme. Let $A = (A_1, A_2)$ be a PPT adversary. We define the advantage of A to distinguish the ciphertexts of two selected equal-length plaintexts as

$$\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},A}(\kappa) := 2 \cdot \Pr\Big[\mathbf{Exp}^{\text{IND-CPA}}_{\mathsf{HE},A}(\kappa)\Big] - 1,$$

where $\mathbf{Exp}^{\text{IND-CPA}}_{\mathsf{HE},A}(\kappa)$ is defined in Experiment 1.5. A is valid, if $|m_0| = |m_1|$. HE is called *IND-CPA secure* (for "indistinguishable under chosen-plaintext attacks") *with access to the evaluation key*, if for every valid A the advantage $\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},A}(\kappa)$ is negligible in κ.

```
$(pk, sk, ek) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$
$(m_0, m_1, state) \leftarrow A_1(1^\kappa, pk, ek)$
$b \leftarrow \{0,1\}$
$c^* \leftarrow \mathsf{HE.Enc}(pk, m_b)$
$b^* \leftarrow A_2(1^\kappa, c^*, state)$
if $b = b^*$ then
    return 1
return 0
```

Experiment 1.5. The public-key IND-CPA experiment of encryption scheme HE with adversary $A = (A_1, A_2)$. A chooses two equal-length messages and guesses afterwards which of the two is the plaintext of the given challenge cipertext c^*. Note that A gets the public and evaluation key beforehand.

Definition 1.11 (Circuit Privacy)**.** Let HE be an N-key $\mathcal{C}$-homomorphic encryption scheme. HE is said to have *perfect (statistical) circuit privacy* for a class of circuits $\mathcal{C}_p \subseteq \mathcal{C}$, if for all $C_1, C_2 \in \mathcal{C}_p$, all tuples $\{(pk_i, sk_i, ek_i)\}_{i \in \{1,\dots,k\}}$, with $k \leq N$ in the support of $\mathsf{HE.Keygen}(1^\kappa)$, all plaintexts $m_1, \dots, m_k \in \Lambda$ and all ciphertexts $c_1, \dots, c_k \in R$, such that c_i is in the support of $\mathsf{HE.Enc}(pk_i, m_i)$, if $C_1(m_1, \dots, m_k) = C_2(m_1, \dots, m_k)$ then the following distributions are perfectly (statistically) indistinguishable:

$$\{\mathsf{HE.Eval}(C_1, (c_1, ek_1), \dots, (c_k, ek_k))\} \text{ and } \{\mathsf{HE.Eval}(C_2, (c_1, ek_1), \dots, (c_k, ek_k))\}.$$

Moreover, it is said to have *computational circuit privacy* for $\mathcal{C}_p$ if the distributions are computationally indistinguishable, even when given the secret keys $(sk_i)_i$.

1.6.1 A Multikey FHE-scheme based on NTRU

We describe a scale-invariant version on the multikey homomorphic encryption scheme of López-Alt, Tromer, and Vaikuntanathan [LTV12], by deducing the feasibility of multikey computations for the LHE' scheme of [B+13]. So note, that this presentation follows mostly the paper of Bos et al. [B+13], with the exception of deducing the multikey noise bounds, similar to those of [LTV12].

The scheme is based on a variant of NTRU due to Stehlé and Steinfeld [SS11] and is relative to the hardness of the Ring Learning with Errors (RLWE) problem as described in Lyubashevsky, Peikert, and Regev [LPR10] and a the non-standard DSPR (Decisional Small Polynomial Ratio) assumption, due to [LTV12]. It features scale-invariance and therefore avoids modulo switching, which makes it much easier to handle. This property is due to Brakerski [B12], and was adapted to the NTRU scheme in [B+13], albeit without taking care of the multikey property of the original scheme.

Encryption schemes based on NTRU use the ring $R := \mathbb{Z}[X]/\Phi_d(X)$, for a $d \in \mathbb{N}_{\geq 1}$, as in Stehlé and Steinfeld [SS11]. Here, $\Phi_d(X)$ denotes d-th cyclotomic polynomial and is of degree $n = \varphi(d)$ (Euler's totient function). The crucial property of Φ_d is that it is monic, i. e., the coefficient of the largest degree is 1, and it is irreducible, i. e., it cannot be written as a product of two polynomials of positive degree. The scheme has message space $\Lambda = R/tR$ and ciphertext space R, where ciphertext are represented modulo q as described below. The possibility of a plaintext modulus $t < q$ of size larger than 2 as in [LTV12] is a generalization due to [B+13].

Notation. We define for $a \in R$ the maximum norm $\|a\|$ as the maximum over the polynomial coefficients, which is well defined. Moreover, we have the following expansion constant associated to R:

$$\delta := \sup_{a,b \in R} \left(\frac{\|ab\|}{\|a\| \|b\|} \right),$$

which is used for our analysis. See [LM06] for bounds on that measure.

Recall that, for a probability distribution χ on R, we write $a \leftarrow \chi$ to indicate that a is the result of sampling from χ. The distribution is said to be B-bounded, $B > 0$, if any $a \leftarrow \chi$ have $\|a\| \in (-B, B)$, cf. [BGV12].

The encryption works on a representation $[a]_q$ of integers in $a \in \mathbb{Z}/q\mathbb{Z}$ as elements in $(-\frac{q}{2}, \frac{q}{2}]$. The corresponding map $[\cdot]_q \colon \mathbb{Z} \to (-\frac{q}{2}, \frac{q}{2}] \cap \mathbb{Z}$ extends to elements of R via an application to the polynomial coefficients, and furthermore to vectors of elements in R. The more common modulo arithmetic $\mathbb{Z} \to [0, q) \cap \mathbb{Z}$ is denoted as $r_q(\cdot)$, to avoid confusion. Moreover, we define $\Delta := \lfloor q/t \rfloor$ and note that $\Delta t = q - r_t(q)$.

The scheme uses key switching [BGV12], a generalized version of relinearization from [BV11a], to remove multiplicities of secret keys upon decryption. Without this method, the size of the ciphertext after evaluation, and the product of secret keys for decryption would depend on the circuit. For instance, already in non-multikey schemes, the square of the secret key f^2 would be needed to decrypt a product after multiplication, if no key switching would be performed to transform the ciphertext back to a cyphertext encrypted under f. For key switching to work, we introduce two functions from [B12], which we denote by $D_{q,w}$ and $P_{q,w}$, for a fixed $w \in \mathbb{N}_{\geq 2}$. For these, let $[a]_q$ be the modulo q representation of $a \in R$ as defined above. We can represent a as a sum $\sum_{j=0}^{\ell_{w,q}-1} [a_j]_w w^j$, with $\ell_{w,q} = \lfloor \log_w q \rfloor + 2$. Then the *decomposition map* $D_{q,w} \colon R \to R^{\ell_{w,q}}$ maps $a \mapsto ([a_j]_w)_{j=0}^{\ell_{w,q}-1}$, whereas for the *power-of-w map* $P_{q,w} \colon R \to R^{\ell_{w,q}}$, we have $a \mapsto ([aw^j]_q)_{j=0}^{\ell_{w,q}-1}$. The salient feature of these maps is that for $a, b \in R$, we have

$$\langle D_{q,w}(a), P_{q,w}(b) \rangle = a \cdot b \pmod{q}.$$

The scheme $\mathsf{HE} := \mathsf{HE}^{(N)} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ works as follows:

- $\mathsf{HE.Keygen}(1^\kappa)$: Choose parameters d, q, t, χ_{key}, χ_{err}, w according to κ, where χ_{key} and χ_{err} are B_{key}-bounded and B_{err}-bounded distributions, respectively. Sample polynomials f', $g \leftarrow \chi_{\text{key}}$ and let $f = [tf' + 1]_q$. We ensure that f is invertible modulo q, by resampling f', if this is not the case. Note that we assume that this does not change the distribution much. Set $h = [tgf^{-1}]_q$. Compute

$$\boldsymbol{\gamma} = [P_{q,w}(f) + \boldsymbol{e} + h\boldsymbol{s}]_q, \quad \text{with } \boldsymbol{e}, \boldsymbol{s} \leftarrow \chi_{\text{err}}^{\ell_{w,q}}.$$

  Output $(pk, sk, ek) = (h, f, \boldsymbol{\gamma})$.

- $\mathsf{HE.Enc}(pk, m)$: To encrypt a message $m + tR$, represented by $[m]_t$, sample $(e, s) \leftarrow \chi_{\text{err}}$ and output $c = [\Delta [m]_t + e + hs]_q$.

- $\mathsf{HE.Dec}(f_1, \ldots, f_l, c)$: Let $l \leq N$. Compute $m = \left[\lfloor t/q \cdot [f_1 \cdots f_l c]_q \rceil \right]_t$.

- $\mathsf{HE.Eval}(C, (c_1, ek_1), \ldots, (c_k, ek_k))$: To evaluate circuit C on $c_1, \ldots, c_k$, we first set $K_i = \{ek_i\}$ and go through the Add and Mult gates of C. During the evaluation, we keep track of the key sets, by passing the resulting key set to the input of the next gate, calling the helper functions defined below.

We use the following helper functions:

- $\mathsf{HE.Add}(K_1, K_2, c_1, c_2)$: Compute $c = [c_1 + c_2]_q$ and output $(c, K_1 \cup K_2)$.

- $\mathsf{HE.KeySwitch}(c, \boldsymbol{k})$: Output $[\langle D_{q,w}(c), \boldsymbol{k}\rangle]_q$.
- $\mathsf{HE.Mult}(K_1, K_2, c_1, c_2)$: Compute $c_{(0)} = [\lfloor t/q \cdot c_1 c_2 \rceil]_q$, and iteratively calculate $c_{(i)} = \mathsf{HE.KeySwitch}\Big(c_{(i-1)}, \boldsymbol{\gamma}\Big)$, for any $\boldsymbol{\gamma} \in K_1 \cap K_2$. Output $(c_{(|K_1 \cap K_2|)}, K_1 \cup K_2)$.

The following lemma shows a condition on the noise of the scheme which would still allow for the correct decryption, together with an argument showing the correctness for freshly encrypted ciphertexts. Here, B_{key} and B_{err} are bounds on the distribution of the key and error term, as in the definition of Keygen.

Lemma 1.3 ([B+13, Lemma 1 and 2])**.** *Let c, f, $m \in R$. If there exists a $v \in R$, such that*

$$fc = \Delta[m]_t + v \pmod q \text{ and } \|v\| < (\Delta - r_t(q))/2,$$

then $\mathsf{HE.Dec}(f, c) = [m]_t$. If $(h, f, \boldsymbol{\gamma}) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ and $c \leftarrow \mathsf{HE.Enc}(h, m)$, then there is a $v \in R$, such that

$$fc = \Delta[m]_t + v \pmod q \text{ and } \|v\| < \delta t B_{\text{key}}\Big(2B_{\text{err}} + \frac{1}{2} r_t(q)\Big).$$

This implies correct decryption for $\Delta > \delta t B_{\text{key}}(4B_{\text{err}} + r_t(q)) + r_t(q)$.

In the following we will analyze the noise growth of the three helper functions used during circuit evaluation, to ensure correct decryption afterwards. Following [B+13], we call $[v]_q$, for $v \in R$ as in the previous lemma, the *inherent noise term* of c.

Addition. Assume that $f_{(i)} c_i = \Delta[m_i]_t + v_i \pmod q$, for $i = 1, 2$, where $f_{(i)}$ is a product $f_1 \cdots f_k \cdot f_{i,1} \cdots f_{i,l_i}$ of $k + l_i$ secret keys f_j. Let the indices $1, \ldots, k$ be of keys, contained in both $f_{(1)}$ and $f_{(2)}$. We define $f = f_1 \cdots f_k$, $\tilde{f}_i = f_{i,1} \cdots f_{i,l_i}$ and $\tilde{f} = f \tilde{f}_1 \tilde{f}_2$, where the latter is the product of the $f_{(1)}$ and $f_{(2)}$ without duplicates. We have

$$\begin{aligned}
\tilde{f}[c_1 + c_2]_q &= \tilde{f} c_1 + \tilde{f} c_2 \\
&= \tilde{f}_2(\Delta[m_1]_t + v_1) + \tilde{f}_1(\Delta[m_2]_t + v_2) \\
&= \Delta \prod_{m=1}^{l_2} \Big(1 + t f'_{2,m}\Big)[m_1]_t + \Delta \prod_{m=1}^{l_1} \Big(1 + t f'_{1,m}\Big)[m_2]_t + \tilde{f}_2 v_1 + \tilde{f}_1 v_2 \\
&= \Delta([m_1]_t + [m_2]_t) + p_2 \cdot \Delta[m_1]_t + p_1 \cdot \Delta[m_2]_t + \tilde{f}_2 v_1 + \tilde{f}_1 v_2 \pmod q,
\end{aligned}$$

with p_1, $p_2 \in R$ such that $\|p_i\| = \|\tilde{f}_i - 1\| < \|\tilde{f}_i\| < \delta^{l_i - 1}(t B_{\text{key}})^{l_i}$. Using $[m_1]_t + [m_2]_t = [m_1 + m_2]_t + t r_{\text{add}}$ with $\|r_{\text{add}}\| \leq 1$, this leads to:

$$\begin{aligned}
\tilde{f}[c_1 + c_2]_q = \Delta[m_1 + m_2]_t + v \pmod q, \text{ where} \\
v = p_2 \cdot \Delta[m_1]_t + p_1 \cdot \Delta[m_2]_t + \tilde{f}_2 v_1 + \tilde{f}_1 v_2 + \Delta t r_{\text{add}}
\end{aligned}$$

and

$$
\begin{aligned}
\|v\| &< \Delta(\|p_2[m_1]_t\| + \|p_1[m_2]_t\|) + \|\tilde{f}_2 v_1\| + \|\tilde{f}_1 v_2\| + r_t(q) \\
&< \delta \frac{r_t(q)}{2}(\|p_2\| + \|p_1\|) + (\delta t B_{\text{key}})^{l_2}\|v_1\| + (\delta t B_{\text{key}})^{l_1}\|v_2\| + r_t(q) \\
&< \frac{r_t(q)}{2}\Big((\delta t B_{\text{key}})^{l_1} + (\delta t B_{\text{key}})^{l_2}\Big) + (\delta t B_{\text{key}})^{l_2}\|v_1\| + (\delta t B_{\text{key}})^{l_1}\|v_2\| + r_t(q).
\end{aligned}
$$

Key Switching. Let $i \in \mathbb{N}_{\geq 1}$. We write this a bit more general than needed for our purposes as in our case $i = 1$ would suffice, but it is an interesting side note that the noise bound is independent of i. Let $(h, f, \boldsymbol{\gamma}) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ and f_K a product of $l \in \mathbb{N}$ secret keys. Let $f^{i+1} f_K \tilde{c} = \Delta[m]_t + \tilde{v} \pmod q$ and $c = \mathsf{HE.KeySwitch}(\tilde{c}, \boldsymbol{k_i})$, where

$$\boldsymbol{k_i} = \Big[P_{q,w}(f^i) + \boldsymbol{e} + h\boldsymbol{s}\Big]_q \in R^{\ell_{w,q}}.$$

Note that $\boldsymbol{\gamma} = \boldsymbol{k_1}$. We obtain c, such that

$$
\begin{aligned}
f f_K c &= f f_K \Big[\langle D_{q,w}(\tilde{c}), P_{q,w}(f^i) + \boldsymbol{e} + h\boldsymbol{s}\rangle\Big]_q \\
&= f^{i+1} f_K \tilde{c} + f f_K \langle D_{q,w}(\tilde{c}), \boldsymbol{e}\rangle + f_K t g \langle D_{q,w}(\tilde{c}), \boldsymbol{s}\rangle \\
&= \Delta[m]_t + v \pmod q,
\end{aligned}
$$

where $v = \tilde{v} + f f_K \langle D_{q,w}(\tilde{c}), \boldsymbol{e}\rangle + f_K t g \langle D_{q,w}(\tilde{c}), \boldsymbol{s}\rangle$.

Therefore, using $\|\langle D_{q,w}(\tilde{c}), \boldsymbol{a}\rangle\| = \|\sum_{j=0}^{\ell_{w,q}-1} [\tilde{c}_i]_w a_i\| \leq \delta \ell_{w,q} \frac{w}{2} \cdot B_{\text{err}}$ for $\boldsymbol{a} \in \{\boldsymbol{e}, \boldsymbol{s}\}$, we obtain:

$$
\begin{aligned}
\|v\| &< \|\tilde{v}\| + (\|f\| + t B_{\text{key}})\|f_K\|\delta \cdot \delta^2 \ell_{w,q} \frac{w}{2} B_{\text{err}} \\
&< \|\tilde{v}\| + \|f_K\|\delta \cdot \delta^2 \ell_{w,q} w t B_{\text{err}} B_{\text{key}},
\end{aligned}
$$

where the factor $\|f_K\|\delta$ can be left out, if $f_K = 1$, as in [B+13, Lemma 5]. Otherwise, we can use $\|f_K\| \leq \delta^{l-1}(t B_{\text{key}})^l$. We define $r_{ks} = \delta^2 \ell_{w,q} w t B_{\text{err}} B_{\text{key}}$.

Multiplication. For multiplication, assume that $f_{(i)} c_i = \Delta[m_i]_t + v_i + q r_i$, as above. Then we have $c_{(0)} = [\lfloor t/q \cdot c_1 c_2 \rceil]_q$. Using the same calculation as in the proof for [B+13, Theorem 6], we obtain:

$$
\begin{aligned}
f_{(1)} f_{(2)} c_{(0)} &= \Delta[m_1 m_2]_t + v_{(0)} \pmod q, \text{ where} \\
v_{(0)} &= [m_1]_t v_2 + [m_2]_t v_1 + t(v_1 r_2 + v_2 r_1) \\
&\quad - r_t(q)([m_1]_t r_2 + [m_2]_t r_1 + r_m) + r_v + r_r - r_a, \text{ with} \\
\|r_m\| &< \frac{1}{2}\delta t, \\
\|r_v\| &\leq \frac{1}{2}\delta \min_i \|v_i\|, \\
\|r_r\| &< \frac{1}{2} + r_t(q)\delta\left(\frac{t}{4} + \frac{1}{2} + \frac{1}{2}\min_i \|v_i\|\right), \text{ and}
\end{aligned}
$$

$$\|[m_1]_t v_2 + [m_2]_t v_1\| \le \delta \frac{t}{2}(\|v_1\| + \|v_2\|).$$

However, we get different bounds on parts of the equation depending on $\|f_{(1)}\|$ and $\|f_{(2)}\|$. As $r_a = f_{(1)} f_{(2)} \frac{t}{q} c_1 c_2 - f_{(1)} f_{(2)} \lfloor \frac{t}{q} c_1 c_2 \rceil$, we have $\|r_a\| \le \delta^2 \|f_{(1)}\| \cdot \|f_{(2)}\| \cdot \|\frac{t}{q} c_1 c_2 - \lfloor \frac{t}{q} c_1 c_2 \rceil\| \le \frac{\delta^2}{2} \|f_{(1)}\| \cdot \|f_{(2)}\|$. Moreover $\|r_i\| < \frac{1}{2}\delta\|f_{(i)}\| + 1$. Using these inequalities and $\|f_{(i)}\| < \delta^{k+l_i-1}(tB_{\text{key}})^{k+l_i}$, we obtain:

$$\begin{aligned}
\|v_{(0)}\| &< \frac{\delta t}{2}\Big((\delta t B_{\text{key}})^{k+l_2} + 3\Big)\|v_1\| + \frac{\delta t}{2}\Big((\delta t B_{\text{key}})^{k+l_1} + 3\Big)\|v_2\| \\
&\quad + r_t(q)\delta\frac{t}{4}\Big((\delta t B_{\text{key}})^{k+l_1} + (\delta t B_{\text{key}})^{k+l_2} + 6\Big) + \frac{1}{2}\delta \min_i \|v_i\| \\
&\quad + \frac{1}{2} + r_t(q)\delta\bigg(\frac{t}{4} + \frac{1}{2} + \frac{1}{2}\min_i\|v_i\|\bigg) + \frac{1}{2}(\delta t B_{\text{key}})^{2k+l_1+l_2} \\
&= \frac{\delta t}{2}\Big((\delta t B_{\text{key}})^{k+l_2} + 3\Big)\|v_1\| + \frac{\delta t}{2}\Big((\delta t B_{\text{key}})^{k+l_1} + 3\Big)\|v_2\| \\
&\quad + \frac{\delta}{2}\min_i\|v_i\|(r_t(q) + 1) + r_t(q)\delta\frac{t}{4}\Big((\delta t B_{\text{key}})^{k+l_1} + (\delta t B_{\text{key}})^{k+l_2}\Big) \\
&\quad + \frac{1}{2}\Big(1 + r_t(q)\delta 4t + (\delta t B_{\text{key}})^{2k+l_1+l_2}\Big).
\end{aligned} \tag{1.3}$$

For the key switching step, let $m = |K_1 \cup K_2| = k + l_1 + l_2$, $f_1, \dots, f_k$ be the secret keys corresponding to $K_1 \cap K_2$ and f the product of all m secret keys. Iteratively calculating $c_{(i)} = \mathsf{HE.KeySwitch}\big(c_{(i-1)}, \gamma\big)$ for any $\gamma \in K_1 \cap K_2$, results in a $c_{(k)}$, with $f c_{(k)} = \Delta[m_1 m_2]_t + v_{(k)} \mod q$, such that

$$\begin{aligned}
\|v_{(k)}\| &< \|v_{(0)}\| + (\sum_{j=2}^{k+1}\|f_j \cdots f_k\|) \cdot \|f\| \cdot \delta^2 r_{ks} \\
&< \|v_{(0)}\| + (\sum_{j=2}^{k+1} \delta^{k-j}(tB_{\text{key}})^{k-j+1}) \cdot \delta^{m-1}(tB_{\text{key}})^m \cdot \delta^2 r_{ks} \\
&= \|v_{(0)}\| + (\sum_{j=0}^{k-1}(\delta t B_{\text{key}})^j) \cdot (\delta t B_{\text{key}})^m \cdot r_{ks} \\
&= \|v_{(0)}\| + \frac{1 - (\delta t B_{\text{key}})^k}{1 - \delta t B_{\text{key}}} \cdot (\delta t B_{\text{key}})^m \cdot r_{ks} \\
&< \|v_{(0)}\| + (\delta t B_{\text{key}})^{2k+l_1+l_2} \cdot r_{ks}
\end{aligned}$$

Remark 1.7. Note that for $k = 1$ and $l_1 = l_2 = 0$, i. e., the single-key case, we get the same bounds as in [B+13], with the exception, that we have an additional additive $r_t(q)$ for addition, and have slightly different constants for the multiplication step, cf. [B+13, Lemma 7]. (Using $\frac{7t+2}{2} \ge 4t$, we have a 3 instead of a 2 in the first line of (1.3), and a 4 instead of a 3 in its last line.)

Correctness. To guarantee the correctness of the scheme for at least L levels of operations, we put ourselves in the worst case scenario, where we have only multiplication gates with the maximum number N of keys involved. Moreover, we assume the inherent noise terms of the values for both gate inputs are approximately the same.

Theorem 1.2. *The scheme can correctly evaluate circuits of depth L, with inherent noise terms $\leq V$, arranged in a leveled binary tree, if*

$$\Delta > 2(C_1^L V + L C_1^{L-1} C_2) + r_t(q),$$

where C_1, C_2 are given as follows

$$C_1 = 2\delta t(\delta t B_{\text{key}})^N, \quad C_2 = 2(\delta t B_{\text{key}})^{2N+1}\delta \ell_{w,q} w B_{\text{err}}.$$

Proof. We first note that the noise terms for multiplication dominate the ones for addition. We summarize the noise terms for multiplication as

$$\begin{aligned}
v_{\text{mult}} &< \tilde{C}_1 V + \tilde{C}_2, \text{ where} \\
\tilde{C}_1 &= \frac{\delta}{2}\Big(t(\delta t B_{\text{key}})^{k+l_2} + t(\delta t B_{\text{key}})^{k+l_1} + 6t + r_t(q) + 1\Big), \\
&< \delta t(\delta t B_{\text{key}})^N + 3\delta t + \frac{1}{2}\delta r_t(q) + \frac{\delta}{2} \\
&< 2\delta t(\delta t B_{\text{key}})^N, \qquad \text{for } \delta \text{ or } B_{\text{key}} \geq 2, \\
\tilde{C}_2 &= r_t(q)\delta \frac{t}{4}\Big((\delta t B_{\text{key}})^{k+l_1} + (\delta t B_{\text{key}})^{k+l_2}\Big) + \frac{1}{2} + 2r_t(q)\delta t \\
&\quad + (\delta t B_{\text{key}})^{2k+l_1+l_2}\left(\frac{1}{2} + \delta^2 \ell_{w,q} w t B_{\text{key}} B_{\text{err}}\right) \\
&< r_t(q)\delta \frac{t}{2}(\delta t B_{\text{key}})^N + \frac{1}{2} + 2r_t(q)\delta t + (\delta t B_{\text{key}})^{2N}\left(\frac{1}{2} + \delta^2 \ell_{w,q} w t B_{\text{key}} B_{\text{err}}\right) \\
&< 2(\delta t B_{\text{key}})^{2N}\delta^2 \ell_{w,q} w t B_{\text{key}} B_{\text{err}}, \qquad \text{for } \delta, B_{\text{err}} \text{ or } B_{\text{key}} \geq 2.
\end{aligned}$$

By an iteration over L levels, and Lemma 1.3, we obtain the bound as stated. □

Security. For the security of the scheme, we need the following hardness assumption.

Definition 1.12 (RLWE and DSPR)**.** Let $\kappa \in \mathbb{N}$ be the security parameter, and d, $q \in \mathbb{N}$ be integers and χ a distribution over R/qR, all depending on κ.

1. The (Decision-)RLWE$_{d,q,\chi}$ (Ring Learning with Errors) problem is to distinguish the distributions of pairs $\{(a, u)\}$ and $\{(a, a \cdot s + e)\}$, where $a, u, s \leftarrow R/qR$ are drawn uniformly at random, s remains fixed for all samples, and $e \leftarrow \chi$. See also [B+13, Definition 1], [LPR10].

2. Let t be invertible in R/qR, $y_i \in R/qR$ and $z_i = -y_i t^{-1} \pmod{q}$ for $i = 1, 2$. The DSPR$_{d,q,\chi}$ (Decisional Small Polynomial Ratio) problem is to distinguish $\{u\}$ from $\{a/b\}$, where $u \leftarrow R/qR$ is drawn uniformly at random and $a \leftarrow y_1 + t\chi_{z_1}$, $b \leftarrow y_2 + t\chi_{z_1}$. See also [B+13, Definition 2], [LTV12, Definition 3.4].

Theorem 1.3 ([LTV12, Lemma 3.6], [B$^+$13, Theorem 8])**.** *For $N \in \mathbb{N}_{\geq 1}$, the parameters d, q and the distributions χ_{key}, χ_{err} as above, $\mathsf{HE}^{(N)}$ is IND-CPA secure under the assumption that the RLWE$_{d,q,\chi_{\text{err}}}$ and the DSPR$_{d,q,\chi_{\text{key}}}$ problems are hard, and that the IND-CPA security is preserved even when the evaluation keys are given to the adversary.*

Fully Homomorphic Scheme. By showing that our scheme can evaluate its own decryption circuit and an additional gate, we can use a multikey variant [LTV12, Theorem 4.5] of Gentry's bootstrapping method [G09] to obtain a fully homomorphic scheme. For this, we have to assume a circular security property, namely that the scheme remains secure, even when given an encryption of the bits of the secret keys. For this section, we set $t = 2$.

Lemma 1.4 ([LTV12, Lemma 4.4], [B$^+$13, Lemma 4])**.** *The decryption circuit for $\mathsf{HE}^{(N)}$ can be implemented as a polynomial size circuit of depth $O(\log N(\log\log q + \log d))$ over $\mathbb{F}_2$.*

Theorem 1.4 (Multikey fully homomorphic encryption)**.** *Under the assumption, that $\{\mathsf{HE}^{(N)}\}_{N\geq 1}$ remains IND-CPA secure even when the adversary is given the evaluation keys and encryptions of the bits of all secret keys, and for Δ as in Theorem 1.2, with $L = c \cdot \log N(\log\log q + \log d)$, $c \geq 1$, we can find a family of IND-CPA secure multikey fully homomorphic encryption schemes.*

Proof. This is a consequence of the ability to evaluate the decryption circuit due to the choice of Δ and [LTV12, Theorem 4.1], a multikey variant of Gentry's bootstrapping theorem. See also [LTV12, Theorem 4.5]. □

2 Computational Arithmetic Secret Sharing

In this chapter we devise a strongly multiplicative computational secret sharing scheme in two variants. The first of these is an adaption of the schemes by [BC95] and [K94] to a fully homomorphic encryption scheme. They combine space-efficient information dispersal algorithms (i. e., SSS with $\mathcal{B} = \varnothing$) with encryption and a perfectly secure SSS to share its much smaller cryptographic keys. As a side note, we would like to mention that CSS can also be based on so-called all-or-nothing transform as in [RP11].

In the first section we give a detailed construction of our scheme and a proof that it is a secret sharing scheme, in a game-based provable-security framework. In Section 2.1.1 we discuss its arithmetic properties, and give a concrete instance of the scheme by suggesting a version for each of its components. In Section 2.2 we point out how we can base an actively secure multiparty computation protocol on our scheme. For notation, let $R\langle X\rangle$ denote the R-algebra generated on a set X.

2.1 Construction of an Arithmetic CSS Scheme

Let $P = \{P_1, \dots, P_n\}$ be the set of players and $\Gamma = (\mathcal{A}, \mathcal{B})$ an access structure on P, and $\mathcal{C}$ a class of circuits. In the following we construct a $\mathcal{C}$-arithmetic computational secret sharing scheme $\mathsf{CSS}[\mathsf{HE}, \mathsf{IDA}, \mathsf{KSS}] = (\mathsf{Sh}, \mathsf{Rec})$ over an R-algebra Λ with message space $\mathcal{M} = \Lambda$, share spaces $\mathcal{S}_i = \mathcal{S}_i^{\mathrm{ida}} \times R\langle \mathcal{S}_i^{\mathrm{kss}} \rangle$ and access structure Γ, using the following ingredients:

- a multikey $\mathcal{C}$-homomorphic scheme $\mathsf{HE} = (\mathsf{Keygen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ with message space Λ, key space $\mathcal{K}$, and ciphertexts over R,
- a linear information dispersal scheme $\mathsf{IDA} = (\mathsf{IDA.Sh}, \mathsf{IDA.Rec})$ over $R' \in \{R, \Lambda\}$ with message space R', share spaces $\mathcal{S}_i^{\mathrm{ida}}$ and access structure $\Gamma' = (\mathcal{A}, \varnothing)$, and
- a secret sharing scheme $\mathsf{KSS} = (\mathsf{KSS.Sh}, \mathsf{KSS.Rec})$ with message space $\mathcal{K}$, share spaces $\mathcal{S}_i^{\mathrm{kss}}$ and access structure Γ.

Then we have to possible variants of sharing and reconstruction protocols, which differ in the order of operations: In the first variant, the secret is first encrypted and then shared, where in the second, it is first shared and the shares are encrypted afterwards. Note that in the definition of the IDA used for our scheme, $R' = R$ in variant 1 and $R' = \Lambda$ in variant 2. We do the proofs for a public-key version of the encryption scheme, although this is not strictly necessary. All proofs go through for secret-key encryption as well. We decided for this version as most fully homomorphic encryption schemes support public-key encryption anyway. While we can reduce these to the secret-key version by setting as the secret key the pair of both keys, in a concrete run we have security also for the case that a dealer stores its public key insecurely or decides to publish it, for whatever reason.

Distribution Variant 1:

1. Generate a random key $(pk, sk, ek) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ and encrypt the secret $m \in \mathcal{M}$ with HE using pk, i. e., $f \leftarrow \mathsf{HE.Enc}(pk, m)$,
2. Generate shares of the ciphertext by IDA, i. e., $\boldsymbol{f} \leftarrow \mathsf{IDA.Sh}(f)$,
3. Generate shares of the encryption key by KSS, i. e., $\boldsymbol{sk} \leftarrow \mathsf{KSS.Sh}(sk)$,
4. Send $(\boldsymbol{f}_i, \boldsymbol{sk}_i)$ to player P_i.

Reconstruction Variant 1:

Let $(\boldsymbol{f}_i, \boldsymbol{sk}_i)$ be the share of player P_i and $A \in \mathcal{A}$, then

1. reconstruct f from the shares using IDA: $f = \mathsf{IDA.Rec}((\boldsymbol{f}_i)_{P_i \in A})$,
2. reconstruct $sk \in R\langle\mathcal{K}\rangle$ from the second component of the shares using KSS: $sk = \mathsf{KSS.Rec}((\boldsymbol{sk}_i)_{P_i \in A})$,
3. decrypt $m = \mathsf{HE.Dec}(\psi(sk), f)$, which yields the reconstructed secret, where ψ is a function reducing the key information in a way suitable for the encryption scheme HE. So, if the FHE from Section 1.6.1 is used, we have $\psi\colon R\langle\mathcal{K}\rangle \to \mathcal{P}(\mathcal{K})$.

Distribution Variant 2:

1. Generate shares of the secret m by IDA, i. e., $\boldsymbol{m} \leftarrow \mathsf{IDA.Sh}(m)$,
2. Generate keys $(pk, sk, ek) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ and encrypt the share vector $\boldsymbol{m}$ component-wise with HE using pk, i. e., $\boldsymbol{f}_i \leftarrow \mathsf{HE.Enc}(pk, \boldsymbol{m}_i)$,
3. Generate shares of the secret key by KSS, i. e., $\boldsymbol{sk} \leftarrow \mathsf{KSS.Sh}(sk)$,
4. Send $(\boldsymbol{f}_i, \boldsymbol{sk}_i)$ to player P_i.

Reconstruction Variant 2:

Let $(\boldsymbol{f}_i, \boldsymbol{sk}_i)$ be the share of player P_i and $A \in \mathcal{A}$, then

1. reconstruct $sk \in R\langle\mathcal{K}\rangle$ from the second to the last component of the shares using KSS: $sk = \mathsf{KSS.Rec}((\boldsymbol{sk}_i)_{P_i \in A})$,
2. decrypt $\boldsymbol{m}_i = \mathsf{HE.Dec}(\psi(sk), \boldsymbol{f}_i)$, which yields the share vector $\boldsymbol{m}$ of the secret. As in variant 1, ψ is a function reducing the key information in a way suitable for the encryption scheme HE.
3. reconstruct m from the the shares $\boldsymbol{m}$ using IDA: $m = \mathsf{IDA.Rec}((m_i)_{P_i \in A})$. This is the reconstructed secret.

Moreover, for homomorphic operations, we have:

Addition. Given shares $(\boldsymbol{f}_i, \boldsymbol{sk}_i)$ and $(\boldsymbol{f}'_i, \boldsymbol{sk}'_i)$ of secrets m and m' and the corresponding evaluation keys ek_i and ek'_i, we have

$$\mathsf{Add}((\boldsymbol{f}_i, \boldsymbol{sk}_i), (\boldsymbol{f}'_i, \boldsymbol{sk}'_i)) \coloneqq (\mathsf{HE.Eval}(C_{\text{add}}, (\boldsymbol{f}_i, ek_i), (\boldsymbol{f}'_i, ek'_i)), \boldsymbol{sk}_i + \boldsymbol{sk}'_i).$$

This yields a share of $m + m'$.

Constant Multiplication. Given a share $(\boldsymbol{f}_i, \boldsymbol{sk}_i)$ of secret $m \in \mathcal{M}$, the corresponding evaluation key ek_i and a factor $u \in \Lambda$, we have

$$\mathsf{CMult}((\boldsymbol{f}_i, \boldsymbol{sk}_i), u) \coloneqq (\mathsf{HE.Eval}(C_{\text{cmult}}, (\boldsymbol{f}_i, ek_i), u), u \cdot \boldsymbol{sk}_i).$$

This yields a share of $u \cdot m$.

For variant 1, we want the following diagram to be commutative:

$$\begin{tikzcd}
\Lambda^t \arrow[r, "\bigtimes_{i=1}^{t} \mathsf{HE.Enc}(pk_i, \cdot)"] \arrow[d, "C"] & R^t \arrow[r, "\bigtimes_{i=1}^{t} \mathsf{HE.Eval}(\mathsf{IDA.Sh}, \cdot)"] \arrow[d, "\mathsf{HE.Eval}(C, \cdot)"] & (R^n)^t \arrow[d, "\mathsf{HE.Eval}(\mathsf{Lift}(C), \cdot)"] \\
\Lambda & R \arrow[l, "\mathsf{HE.Dec}(sk_1, \dots, sk_t, \cdot)"] & R^n \arrow[l, "\mathsf{HE.Eval}(\mathsf{IDA.Rec}, \cdot)"]
\end{tikzcd}$$

For variant 2, we have the following commutative diagram:

$$\begin{tikzcd}
\Lambda^t \arrow[r, "\bigtimes_{i=1}^{t} \mathsf{IDA.Sh}"] \arrow[d, "C"] & (\Lambda^n)^t \arrow[r, "\bigtimes_{i=1}^{t} \mathsf{HE.Enc}(pk_i, \cdot)^n"] \arrow[d, "\mathsf{Lift}(C)"] & (R^n)^t \arrow[d, "\mathsf{HE.Eval}(\mathsf{Lift}(C), \cdot)"] \\
\Lambda & \Lambda^n \arrow[l, "\mathsf{IDA.Rec}"] & R^n \arrow[l, "\mathsf{HE.Dec}(sk_1, \dots, sk_t, \cdot)^n"]
\end{tikzcd}$$

We compare both variants in Chapter 4. For now, we show that these schemes satisfy the definition of a secret sharing scheme.

Proposition 2.1. *Let* $\mathsf{CSS}[\mathsf{HE}, \mathsf{IDA}, \mathsf{KSS}]$ *be the distribution scheme as defined above. The following holds for both variants:*

1. *Let A be a PPT $\mathcal{B}$-privacy adversary for* CSS*. Then there is a valid PPT adversary B for attacking the IND-CPA security of* HE *and a PPT $\mathcal{B}$-privacy adversary C for* KSS*, such that*

$$\mathbf{Adv}^{\text{priv}}_{\mathsf{CSS},A}(\kappa) \leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},B}(\kappa) + 2 \cdot \mathbf{Adv}^{\text{priv}}_{\mathsf{KSS},C}(\kappa).$$

2. *Let A be an $\mathcal{A}$-reconstruction adversary of* CSS *(with share erasures). Then there are $\mathcal{A}$-reconstruction (erasure) adversaries B and C for* KSS *and* IDA*, resp., such that*
$$\mathbf{Adv}^{\mathrm{rec}}_{\mathsf{CSS},A}(\kappa) \leq \mathbf{Adv}^{\mathrm{rec}}_{\mathsf{KSS},B}(\kappa) + \mathbf{Adv}^{\mathrm{rec}}_{\mathsf{IDA},C}(\kappa).$$
Moreover, when A is PPT, then B and C are PPT as well.

Hence, CSS *is a computational secret sharing scheme with access structure $\Gamma = (\mathcal{A}, \mathcal{B})$. Moreover, if* IDA *and* KSS *have (computationally) robust reconstructability, then so has* CSS.

Proof. This is a modification of the corresponding proof in [BR07, Theorem 4 and 5]. We first show the statements for *variant 1*:

1. Let game G_1^A be the priv-experiment of CSS with $\mathcal{B}$-privacy adversary A, so that
$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{CSS},A}(\kappa) = 2 \cdot \Pr\Big[\mathrm{out}(G_1^A) = 1\Big] - 1.$$

   We devise a modification of game G_1^A as follows: let (pk, sk, ek) denote the keys used to encrypt the secret $m \in \mathcal{M}$ by $f \leftarrow \mathsf{HE.Enc}(pk, m)$. Instead of sharing sk via $\mathsf{KSS.Sh}$, we independently draw an additional key $(pk', sk', ek') \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ of the same length and share the secret key via $\mathsf{KSS.Sh}(sk')$. We call this game G_2^A and note that any corruption oracle calls send the part of the key share generated by sk'. In effect, while the secret is encrypted with sk, the shares of the key contain an independent key sk'.

   Our task is now to construct a $\mathcal{B}$-privacy adversary C for KSS to distinguish games G_1^A and G_2^A. Internally, it simulates game G_1^A and receives m_0, m_1 from A as input. It then generates equal-length keys sk_0, sk_1 by $\mathsf{HE.Keygen}(1^\kappa)$, together with pk_0, ek_0, pk_1, ek_1, and sends these out as part of the priv game of KSS. Furthermore, it draws $b \leftarrow \{0,1\}$, encrypts $f \leftarrow \mathsf{HE.Enc}(pk_0, m_b)$ and creates a share vector of f via $\boldsymbol{f} \leftarrow \mathsf{Sh}(f)$. Whenever A uses its corruption oracle on player P_i, C uses $\mathsf{corrupt}(\boldsymbol{sk}, i)$ to obtain $\boldsymbol{sk}_i$—the share of P_i of $sk_{b'}$, $b' \leftarrow \{0,1\}$—and answers A's oracle call by $\boldsymbol{s}_i = (\boldsymbol{f}_i, \boldsymbol{sk}_i)$. At the end, the output of G_1^A is forwarded as the output of C.

   To recognize that C is a distinguisher of games G_1^A and G_2^A, note that if $sk_{b'} = sk_0$ we obtain game 1 and if $sk_{b'} = sk_1$, we have the second game. Hence,
$$\Pr\Big[\mathrm{out}(G_1^A) = 1\Big] - \Pr\Big[\mathrm{out}(G_2^A) = 1\Big] = \mathbf{Adv}^{\mathrm{priv}}_{\mathsf{KSS},C}(\kappa).$$

   Next, we show that a valid PPT adversary B attacking the IND-CPA security of HE can be used to decide game 2, i. e.,
$$2 \cdot \Pr\Big[\mathrm{out}(G_2^A) = 1\Big] - 1 \leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},B}(\kappa).$$

   For this, B behaves as follows: Internally, it simulates game G_2^A and receives m_0, m_1 from A as input, and pk, ek as an input of the game. m_0, m_1 are then send

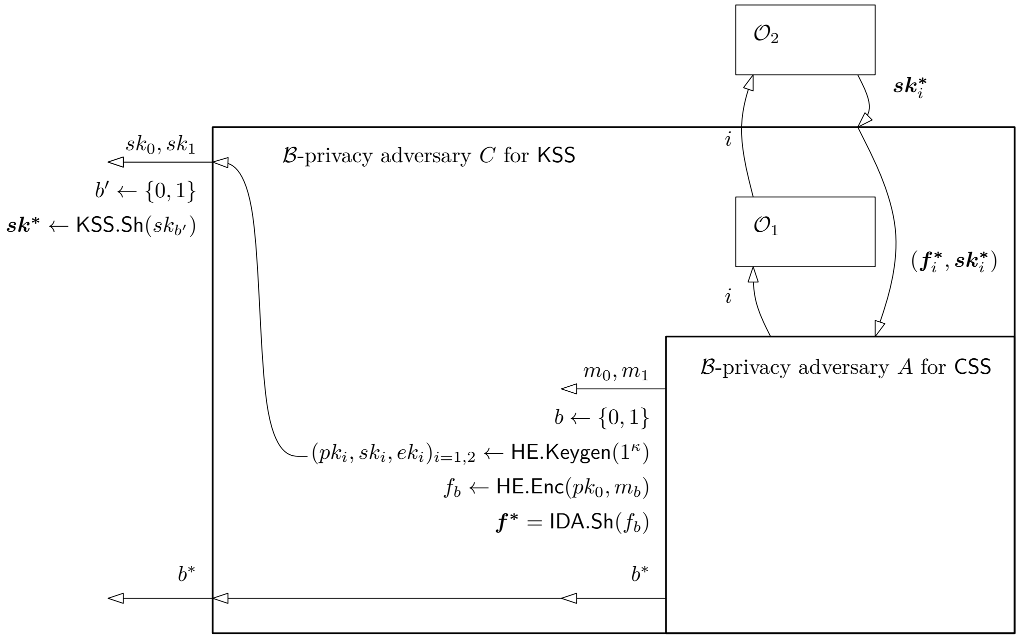
$\mathcal{O}_2$
$\boldsymbol{sk}_i^*$
sk_0, sk_1
$\mathcal{B}$-privacy adversary C for KSS
i
$b' \leftarrow \{0,1\}$
$\boldsymbol{sk}^* \leftarrow \mathsf{KSS.Sh}(sk_{b'})$
$\mathcal{O}_1$
$(\boldsymbol{f}_i^*, \boldsymbol{sk}_i^*)$
i
m_0, m_1
$\mathcal{B}$-privacy adversary A for CSS
$b \leftarrow \{0,1\}$
$(pk_i, sk_i, ek_i)_{i=1,2} \leftarrow \mathsf{HE.Keygen}(1^\kappa)$
$f_b \leftarrow \mathsf{HE.Enc}(pk_0, m_b)$
$\boldsymbol{f}^* = \mathsf{IDA.Sh}(f_b)$
b^*
b^*

Figure 2.1. Reduction step for the indistinguishability of games 1 and 2 in variant 1. Here, $\mathcal{O}_1$ and $\mathcal{O}_2$ represent the corruption oracles of G_1^A and the KSS-priv games, resp.

out as part of the IND-CPA game of HE. Upon retrieval of a ciphertext c^* of m_b, $b \leftarrow \{0,1\}$ under a key pk, it generates $\boldsymbol{c}^* \leftarrow \mathsf{Sh}(c^*)$, draws a key $(pk', sk', ek') \leftarrow \mathsf{HE.Keygen}(1^\kappa)$, and shares sk' as well by $\boldsymbol{sk'} \leftarrow \mathsf{Sh}(sk')$. Now, when A makes use of its corruption oracle $\mathsf{corrupt}(\boldsymbol{s}, i)$, B sends a share $\boldsymbol{s}_i = (\boldsymbol{c}_i^*, \boldsymbol{sk}_i')$. When G_2^A outputs a bit b^*, this is forwarded as the output of B.

This yields

$$\begin{aligned}
\mathbf{Adv}^{\text{priv}}_{\mathsf{CSS},A}(\kappa) &= 2\Big(\Pr\Big[\text{out}(G_1^A) = 1\Big] + \Big(\Pr\Big[\text{out}(G_1^A) = 1\Big] - \Pr\Big[\text{out}(G_2^A) = 1\Big]\Big)\Big) - 1 \\
&= 2 \cdot \Pr\Big[\text{out}(G_1^A) = 1\Big] - 1 + 2 \cdot \mathbf{Adv}^{\text{priv}}_{\mathsf{KSS},C}(\kappa) \\
&\leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},B}(\kappa) + 2 \cdot \mathbf{Adv}^{\text{priv}}_{\mathsf{KSS},C}(\kappa).
\end{aligned}$$

2. Let m be the output of adversary A in the reconstruction game of CSS. Let $(pk, sk, ek) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$ be the keys used to encrypt m into ciphertext $f \leftarrow \mathsf{HE.Enc}(pk, m)$. The corruption oracle outputs $\boldsymbol{s}_i = (\boldsymbol{f}_i, \boldsymbol{sk}_i)$ as usual. Let $\boldsymbol{s'} = (\boldsymbol{f'}, \boldsymbol{sk'}), j$ be A's output and denote by $f' = \mathsf{IDA.Rec}(\boldsymbol{f}_{\overline{T}} \sqcup \boldsymbol{f}'_T, j)$ and $sk' = \mathsf{KSS.Rec}(\boldsymbol{sk}_{\overline{T}} \sqcup \boldsymbol{sk}'_T, j)$. In the case $f = f'$ and $\psi(sk) = \psi(sk')$, the recovery of $\boldsymbol{s}_{\overline{T}} \sqcup \boldsymbol{s}'_T$ yields m. So if E_1 is the event that $f \neq f'$ and E_2 the event that $\psi(sk) \neq \psi(sk')$, we have that

$$\mathbf{Adv}^{\text{rec}}_{\mathsf{CSS},A}(\kappa) \leq \Pr[E_1 \vee E_2] \leq \Pr[E_1] + \Pr[E_2].$$

But note that we have $\Pr[E_1] \leq \mathbf{Adv}^{\mathrm{rec}}_{\mathsf{IDA},B}(\kappa)$ and $\Pr[E_2] \leq \mathbf{Adv}^{\mathrm{rec}}_{\mathsf{KSS},C}(\kappa)$, by definition of IDA and KSS.

Next, we show the analogous statements for *variant 2*:

1. Let game G_1^A be the priv-experiment of CSS with $\mathcal{B}$-privacy adversary A, so that

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{CSS},A}(\kappa) = 2 \cdot \Pr\Big[\mathrm{out}(G_1^A) = 1\Big] - 1.$$

Game G_2^A is defined exactly as in the proof for variant 1. Our task is then to construct a $\mathcal{B}$-privacy adversary C for KSS to distinguish games G_1^A and G_2^A. Internally, it simulates game G_1^A and receives m_0, m_1 from A as input. It then generates equal-length keys sk_0, sk_1 by $\mathsf{HE.Keygen}(1^\kappa)$, together with pk_0, ek_0, pk_1, ek_1, and sends these out as part of the priv game of KSS. Furthermore, it draws $b \leftarrow \{0,1\}$, shares m_b by $\boldsymbol{s} \leftarrow \mathsf{IDA.Sh}(m_b)$ and encrypts the shares component-wise under pk_0: $\boldsymbol{f}_i \leftarrow \mathsf{HE.Enc}(pk_0, \boldsymbol{s}_i)$. The corruption oracle works exactly as in the proof for variant 1.

To recognize that C is a distinguisher of games G_1^A and G_2^A, note that if $sk_{b'} = sk_0$ we obtain game 1 and if $sk_{b'} = sk_1$, we have the second game. Hence,

$$\Pr\Big[\mathrm{out}(G_1^A) = 1\Big] - \Pr\Big[\mathrm{out}(G_2^A) = 1\Big] = \mathbf{Adv}^{\mathrm{priv}}_{\mathsf{KSS},C}(\kappa).$$

Next, we show that a valid PPT adversary B attacking the IND-CPA security of HE can be used to decide game 2, i. e.,

$$2 \cdot \Pr\Big[\mathrm{out}(G_2^A) = 1\Big] - 1 \leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},B}(\kappa).$$

For this, B behaves as follows: Internally, it simulates game G_2^A and receives m_0, m_1 from A as input, and pk, ek as an input of the game. Now, $\boldsymbol{s_0}, \boldsymbol{s_1}$ are generated by IDA, and are sent out as part of the IND-CPA game of HE. Note that these are multiple elements, but by a hybrid argument, this is not a problem. Upon retrieval of a ciphertext vector $\boldsymbol{c}^*$ of $\boldsymbol{s_b}$, $b \leftarrow \{0,1\}$ under a key sk, it draws a key $(pk', sk', ek') \leftarrow \mathsf{HE.Keygen}(1^\kappa)$, and shares sk' by $\boldsymbol{sk'} \leftarrow \mathsf{Sh}(sk')$. Now, when A makes use of its corruption oracle $\mathsf{corrupt}(\boldsymbol{s}, i)$, B sends a share $\boldsymbol{s}_i = (\boldsymbol{c}_i^*, \boldsymbol{sk}'_i)$. When G_2^A outputs a bit b^*, this is forwarded as the output of B.

This yields, as above

$$\mathbf{Adv}^{\mathrm{priv}}_{\mathsf{CSS},A}(\kappa) \leq \mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},B}(\kappa) + 2 \cdot \mathbf{Adv}^{\mathrm{priv}}_{\mathsf{KSS},C}(\kappa).$$

2. This proof is completely analogous to the corresponding proof of variant 1. □

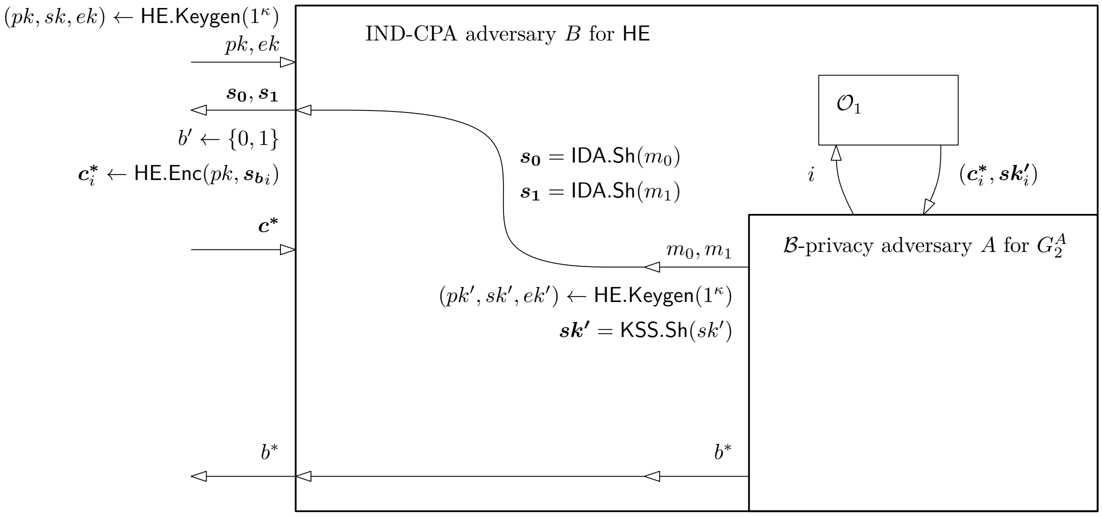
$(pk, sk, ek) \leftarrow \mathsf{HE.Keygen}(1^\kappa)$
pk, ek
IND-CPA adversary B for HE
$\boldsymbol{s_0}, \boldsymbol{s_1}$
$b' \leftarrow \{0,1\}$
$\boldsymbol{c}_i^* \leftarrow \mathsf{HE.Enc}(pk, \boldsymbol{s}_{\boldsymbol{b}_i})$
$\boldsymbol{c}^*$
$\mathcal{O}_1$
$\boldsymbol{s_0} = \mathsf{IDA.Sh}(m_0)$
$\boldsymbol{s_1} = \mathsf{IDA.Sh}(m_1)$
i
$(\boldsymbol{c}_i^*, \boldsymbol{sk}_i')$
m_0, m_1
$\mathcal{B}$-privacy adversary A for G_2^A
$(pk', sk', ek') \leftarrow \mathsf{HE.Keygen}(1^\kappa)$
$\boldsymbol{sk'} = \mathsf{KSS.Sh}(sk')$
b^*
b^*

Figure 2.2. Reduction step for the indistinguishability of IND-CPA in variant 2. Here, $\mathcal{O}_1$ represents the corruption oracle of G_2^A. The figure for variant 1 is obtained similarly.

2.1.1 Arithmetic Properties of the Scheme

To see that CSS is homomorphic, we have to specify a multiplication on the share spaces and a special share function Sh_i' on the share spaces, to fulfill Definition 1.3. For this, we set for $\boldsymbol{s}_i = (\boldsymbol{f}_i, \boldsymbol{sk}_i)$ and $\boldsymbol{s}_i' = (\boldsymbol{f}_i', \boldsymbol{sk}_i')$ in $\mathcal{S}_i^{\text{ida}} \times R\langle\mathcal{S}_i^{\text{kss}}\rangle$ with corresponding evaluation keys ek, ek':

$$\boldsymbol{s}_i \circledast_i \boldsymbol{s}_i' := \big(\mathsf{HE.Eval}\big(C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}_i', ek')\big), \boldsymbol{sk}_i \cdot \boldsymbol{sk}_i'\big).$$

Now for variant 1, we set $\mathsf{Sh}_i' := \mathsf{IDA.Sh} \times \mathsf{KSS.Sh}$, i. e., for $(\boldsymbol{f}, \boldsymbol{sk}) \in \mathcal{S}_i^{\text{ida}} \times R\langle\mathcal{S}_i^{\text{kss}}\rangle$, we have:

$$\mathsf{Sh}_i'((\boldsymbol{f}, \boldsymbol{sk})) = \Big(\mathsf{HE.Eval}(\mathsf{IDA.Sh}, (\boldsymbol{f}, ek))_j, \mathsf{KSS.Sh}(\boldsymbol{sk})_j\Big)_{j=1,\dots,n}.$$

For variant 2, we need to use the exact same definition for Sh_i', as we need a share function on the share spaces. Hence, we cannot use the standard IDA.Sh of variant 2, but have to use IDA.Sh of variant 1. For this, we assume that these are compatible, i. e., the global Rec function of CSS can recover the result nevertheless.

Remark 2.1. If HE is C^{lin}-homomorphic, then CSS is a linear secret sharing scheme. For this to work, we had to impose an additional structure on key share space, which is actually more than necessary for the scheme to work. When we choose as HE our scheme of Section 1.6.1, we only need a set of secret keys to be reconstructed, and we can discard the information, whether there was an addition or multiplication performed. However, if we replaced $R\langle\mathcal{K}\rangle$ and $R\langle\mathcal{S}_i^{\text{kss}}\rangle$ by the power sets $\mathcal{P}(\mathcal{K})$ and $\mathcal{P}(\mathcal{S}_i^{\text{kss}})$, respectively, we would not have a module structure in the usual sense, and hence no linear reconstruction map. Let us nevertheless describe how this relaxation would work.

$(\mathcal{P}(\mathcal{K}), \cup)$ is a bounded semilattice, and can therefore be seen as a commutative monoid, i. e., an abelian group without the property that every element has an additive inverse. Note that $(\mathbb{N}, +)$ is an example of a commutative monoid. Moreover, it is idempotent, which means that for any set $A \in \mathcal{P}(\mathcal{K})$ it holds that $A \cup A = A$.

Analogously, a commutative semiring with 1 is a set R with two binary operations $+$ and $*$, such that $(R, +)$ and $(R, *)$ are commutative monoids, in which the distributivity laws $a * (b + c) = (a * b) + (a * c)$, and $(a + b) * c = (a * c) + (b * c)$, for a, b, $c \in R$ hold, and additionally $0 * a = a * 0 = 0$, for $a \in R$. This is in essence the definition of a commutative ring with 1, without the requirement that $(R, +)$ is a group. An R-module $\mathcal{M}$ over a semiring, or semimodule for short, is a commutative monoid $(M, +)$ with an operation $R \times M \to M$, such that $r(x + y) = rx + ry$, $(r + s)x = rx + sx$, $(rs)x = r(sx)$ and $1x = x$. This is in essence the definition of a module over a ring, without the condition that $(M, +)$ is a group. Note that any commutative monoid is naturally an $\mathbb{N}$-semimodule. Moreover, an R-algebra A over a semiring is an R-semimodule, with the additional R-bilinear multiplication operation.

In this sense Rec would still be a semimodule homomorphism, and it might be an interesting question, whether these properties are sufficient to show that the scheme is fitted for the application of MPC. We will leave this open for now.

Theorem 2.1. *Let* $\mathsf{CSS}[\mathsf{HE}, \mathsf{IDA}, \mathsf{KSS}] = (\mathsf{Sh}, \mathsf{Rec})$ *be the secret sharing scheme as defined above in variant 1 or 2. Let* $\mathcal{C} \subseteq \mathcal{C}_{t \to l}$ *be a class of circuits on* $\mathcal{M}$*. If* IDA *and* KSS *are* $\mathcal{C}$*-arithmetic and* HE *is* $\mathcal{C}$*-homomorphic, then* CSS *is (computationally)* $\mathcal{C}$*-arithmetic. Moreover, if* IDA *and* KSS *are compatible with d-fold multiplications as in Remark 1.3, then so is* CSS*.*

Proof. We check the three conditions of Definition 1.6.

1. The first condition holds, as we have the R-algebra structure enforced on the share spaces by assumption.

2. Instead of showing that for any $\mathcal{C}$-arithmetic $\mathcal{B}$-privacy PPT adversary A of CSS, there is a $\mathcal{B}$-privacy PPT adversary B of CSS, such that

$$\mathbf{Adv}_{\mathsf{CSS},A}^{\mathcal{C}\text{-priv}}(\kappa) \leq \mathbf{Adv}_{\mathsf{CSS},B}^{\text{priv}}(\kappa),$$

we show that there is a valid PPT adversary B for attacking the IND-CPA security of HE and a PPT $\mathcal{C}$-arithmetic $\mathcal{B}$-privacy adversary C for KSS such that

$$\mathbf{Adv}_{\mathsf{CSS},A}^{\mathcal{C}\text{-priv}}(\kappa) \leq t \cdot \mathbf{Adv}_{\mathsf{HE},B}^{\text{IND-CPA}}(\kappa) + 2 \cdot \mathbf{Adv}_{\mathsf{KSS},C}^{\mathcal{C}\text{-priv}}(\kappa)$$

directly, as this is also negligible by assumption. We can reduce the IND-CPA property of HE to the corresponding circuit game by a hybrid argument as in [G09, Chapter 2, p. 32]. With this, the proof is very analog to the proof of Proposition 2.1, and is therefore omitted here.

3. By using Lemma 1.2 it suffices to show strong multiplicativity of Σ, if $\mathcal{C}$ contains a circuit with a multiplication gate. That is, we have to show that for all $j \in \{0, \dots, n\}$, m, $m' \in \mathcal{M}$ with $\boldsymbol{s} \leftarrow \mathsf{Sh}(m)$, $\boldsymbol{s'} \leftarrow \mathsf{Sh}(m')$, it holds:

$$m \cdot m' = \mathsf{Rec}(\boldsymbol{s} * \boldsymbol{s'}, j), \text{ where } \boldsymbol{s} * \boldsymbol{s'} := \sum_i \mathsf{Sh}'_i(\boldsymbol{s}_i \circledast_i \boldsymbol{s}'_i).$$

We do this first for variant 1: Let $\boldsymbol{s}_i = (\boldsymbol{f}_i, \boldsymbol{sk}_i)$ and $\boldsymbol{s}'_i = (\boldsymbol{f}'_i, \boldsymbol{sk}'_i)$ in $\mathcal{S}_i^{\text{ida}} \times R\langle \mathcal{S}_i^{\text{kss}} \rangle$ with corresponding evaluation keys ek, ek'. Then:

$$\begin{aligned}
\mathsf{Rec}(\boldsymbol{s} * \boldsymbol{s'}, j) &= \sum_i \mathsf{Rec}(\mathsf{Sh}'_i((\boldsymbol{f}, \boldsymbol{sk})_i \circledast_i (\boldsymbol{f'}, \boldsymbol{sk'})_i), j) \\
&= \sum_i \mathsf{Rec}(\mathsf{Sh}'_i((\mathsf{HE.Eval}(C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek')), \boldsymbol{sk}_i \cdot \boldsymbol{sk}'_i)), j) \\
&= \sum_i \mathsf{Rec}((\mathsf{HE.Eval}(\mathsf{IDA.Sh} \circ C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek')), \\
&\qquad\quad \mathsf{KSS.Sh}(\boldsymbol{sk}_i \cdot \boldsymbol{sk}'_i)), j) \\
&= \sum_i \mathsf{HE.Dec}(\psi(\mathsf{KSS.Rec}(\mathsf{KSS.Sh}(\boldsymbol{sk}_i \cdot \boldsymbol{sk}'_i))), \\
&\qquad\qquad (\mathsf{HE.Eval}(\mathsf{IDA.Rec} \circ \mathsf{IDA.Sh} \circ C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek'))) \\
&= \sum_i \mathsf{HE.Dec}(sk, sk', \mathsf{HE.Eval}(C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek'))) \\
&= m \cdot m'.
\end{aligned}$$

Which is want we wanted to show. For variant 2, the deduction is similar:

$$\begin{aligned}
\mathsf{Rec}(\boldsymbol{s} * \boldsymbol{s'}, j) &= \sum_i \mathsf{Rec}(\mathsf{Sh}'_i((\boldsymbol{f}, \boldsymbol{sk})_i \circledast_i (\boldsymbol{f'}, \boldsymbol{sk'})_i), j) \\
&= \sum_i \mathsf{Rec}(\mathsf{Sh}'_i((\mathsf{HE.Eval}(C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek')), \boldsymbol{sk}_i \cdot \boldsymbol{sk}'_i)), j) \\
&= \sum_i \mathsf{Rec}((\mathsf{HE.Eval}(\mathsf{IDA.Sh} \circ C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek')), \\
&\qquad\quad \mathsf{KSS.Sh}(\boldsymbol{sk}_i \cdot \boldsymbol{sk}'_i)), j) \\
&= \sum_i \mathsf{IDA.Rec}(\mathsf{HE.Dec}(\psi(\mathsf{KSS.Rec}(\mathsf{KSS.Sh}(\boldsymbol{sk}_i \cdot \boldsymbol{sk}'_i))), \\
&\qquad\qquad \mathsf{HE.Eval}(\mathsf{IDA.Sh} \circ C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek')) \\
&= \sum_i \mathsf{IDA.Rec}(\mathsf{HE.Dec}(sk, sk', \\
&\qquad\qquad \mathsf{HE.Eval}(\mathsf{IDA.Sh} \circ C_{\text{mult}}, (\boldsymbol{f}_i, ek), (\boldsymbol{f}'_i, ek'))))
\end{aligned}$$

Here, we need compatibility with the share function of variant 1, to get

$$\begin{aligned}
&= \sum_i \mathsf{IDA.Rec}((\mathsf{IDA.Sh}(m \cdot m'))_i) \\
&= m \cdot m'.
\end{aligned}$$

This holds, even when the share vector is restricted to a qualified player set in $\mathcal{A}$.

4. For the compatibility with d-fold multiplications the proof proceeds exactly as in the previous item for strong multiplicativity. □

A Concrete Choice. Let us suggest a concrete choice for HE, IDA and KSS, which fulfills the conditions needed in the construction of CSS. For this, let $\mathcal{C}$ be a circuit class on $\mathcal{M}$. Let n be the number of players and $N \in \mathbb{N}_{\geq n}$ and $s \in \mathbb{N}_{\geq 2}$. Let $\Gamma = (\mathcal{A}, \mathcal{B})$ be an access structure.

1. For HE we choose the N-key $\mathcal{C}$-homomorphic encryption scheme of Section 1.6.1. Hence, we set $R := \mathbb{Z}[X]/\Phi_d(X)$ as the ciphertext and the key space of HE, although all elements will be represented modulo q, cf. Section 1.6.1. We set $n := \varphi(d)$ and take $\Lambda := R/tR$ as our message space for some $t < q$. We require that q and t are prime powers. Therefore, we can represent an element of R/qR or R/tR as a vector in $\mathbb{F}_q^n$ and $\mathbb{F}_t^n$, respectively, consisting of the coefficients of a representation of a polynomial in R of degree $\leq n$.

2. For IDA we choose an $(n, s, \mathcal{A})$-arithmetic information dispersal algorithm as in Definition 3.6 over $k = \mathbb{F}_q$ (for variant 1) or $k = \mathbb{F}_t$ (for variant 2), with message space k^n and share spaces $\mathcal{S}_i^{\text{ida}} = k$.

3. For KSS we choose a an (n, s, Γ)-arithmetic secret sharing scheme as in Definition 3.6 over $\mathbb{F}_q$ with message space $\mathbb{F}_q^{\varphi(d)}$ and share spaces $\mathbb{F}_q$.

2.2 Secure Multiparty Computation based on CSS

First note that our CSS scheme is already suitable for the passively secure multiparty computation protocol from Theorem 1.1, as in its proof, nothing about the linearity of the share function is assumed.

In this section, due to time restraints, we only point the reader to the work of Maurer [M03]. In his article, he makes use of so-called replicated secret sharing, also described in Cramer, Damgård, and Ishai [CDI05], to construct an MPC protocol which does not rely on specialized mathematical properties of the scheme, such as linearity of the share map. See also [H$^+$11, Section 3] for an improved and generalized version of the scheme which also exhibits a graceful degradation property.

However, as mentioned in [CDI05] these may inflict an $\binom{n}{t}$ overhead on the communication. Note that a formal proof of the correctness of this argumentation is left as future work.

3 Algebraic Geometric Secret Sharing

In this chapter we want to review the basics of algebraic geometric codes, and its use for the construction of secret sharing schemes, in particular as in Cascudo, Cramer, and Xing [CCX11; CCX12b]. The study of algebraic geometric codes, also called geometric Goppa codes, in the field of secret sharing was initiated by [CC06], which allowed for the construction of linear secret sharing schemes with good parameters and which work over small fields. [D08] is a review article on codes with a focus on this application.

We start with a short overview of the chapter. In Section 3.1 we will introduce some basics from the theory of algebraic function fields, which are used throughout the chapter, including the Riemann–Roch theorem and geometric Goppa codes. Section 3.2 then gives an account on constructions of infinite class field towers, which are used to construct certain codes in [CCX12b]. This is followed by Section 3.3 on Riemann–Roch systems of equations. It formalizes the notion of equation systems on the dimension of the Riemann–Roch space of divisor classes. These are used to obtain divisors which are fitted for the secret sharing application and generalize the previous methods. We show bounds on the so-called torsion limits in Section 3.4. Finally, we are then able to give a construction of an infinite family of secret sharing schemes with interesting arithmetic properties in Section 3.5, based on the results of the previous sections.

The main object of modern coding theory are global function fields. We give the corresponding definitions based on the account of Rosen [R02]. An *algebraic function field* (in one variable) over a field k is a field F, with $k \subseteq F$, and which contains an element x, transcendental over k, such that $F|k(x)$ is a finite algebraic extension. We call k the *constant field* of F. Note that in the case of a finite k, we speak of *global function fields*, as they are—together with algebraic number fields—an instance of so-called *global fields*. As both types of fields share some striking similarities, they are usually considered together in areas such as algebraic number theory. As we concentrate on the case of function fields, we start with an arbitrary constant field, and reduce this generality as needed. Note that the algebraic closure of k in F is finite over k, which is why we usually assume that k is algebraically closed in F. In that case, we say that k is the *exact* or *full constant field* of F.

3.1 Preliminaries

Let F be an algebraic function field over constant field k, which we abbreviate by F/k. A *valuation ring* of F/k is a ring $\mathcal{O}$ such that $k \subsetneq \mathcal{O} \subsetneq F$ and for any $a \in F$, we have that $a \in \mathcal{O}$ or $a^{-1} \in \mathcal{O}$. Valuation rings are local, i. e., they contain a unique maximal ideal. A *prime* in F is then a (discrete) valuation ring $\mathcal{O}$ with maximal ideal $\mathfrak{p}$. As a matter of simplification, we usually address the prime by its maximal ideal $\mathfrak{p}$, as no confusion may arise.

Let κ denote the *residue class field* $\mathcal{O}/\mathfrak{p}$, where $\mathcal{O}$ is the valuation ring associated to $\mathfrak{p}$. The degree $\deg(\mathfrak{p})$ of $\mathfrak{p}$ is then the k-dimension of κ over k, denoted by $[\kappa : k]$ as usual. Any discrete valuation ring has attached to it a discrete *valuation* $v_\mathfrak{p} \colon F \to \mathbb{Z} \cup \{\infty\}$, which is defined via the unique representation of non-zero elements of F as $a = u \cdot t^n$, where t is a generator of $\mathfrak{p}$, $u \in \mathcal{O}^\times$ and $n \in \mathbb{Z}$. Set $v_\mathfrak{p}(a) := n$ in this case, and $v_\mathfrak{p}(0) := \infty$.

A *divisor* D is an element of the free abelian group generated by the primes of F/k, which we also call prime divisors in this context. Denote this group by $\operatorname{Div}(F)$. The coefficients of D are denoted by $v_\mathfrak{p}(D)$. Using this, we define the degree of the divisor D as $\deg(D) = \sum_\mathfrak{p} v_\mathfrak{p}(D) \cdot \deg(\mathfrak{p})$. We denote the set of divisors of degree d by $\operatorname{Div}^{(d)}(F)$.

We can associate to any $f \in F^\times$, the *principal divisor* (f) of f. The corresponding homomorphism $(\cdot) \colon F^\times \to \operatorname{Div}(F)$ is defined as $f \mapsto \sum_\mathfrak{p} v_\mathfrak{p}(f) \cdot \mathfrak{p}$. Its image is called $\operatorname{Prin}(F)$, the group of principal divisors of F. Moreover, two divisors are said to be *linearly equivalent*, if their difference is a principal divisor. Using this equivalence relation, the *divisor class group* $\operatorname{Cl}(F) := \operatorname{Div}(F)/\operatorname{Prin}(F)$ is obtained. As the degree of a principal divisor is zero, the degree function factors through $\operatorname{Cl}(F)$. We denote its kernel by J_F and call it the *zero divisor class group*. The name is motivated by its connection to the Jacobian variety of a smooth curve, when interpreted geometrically. We will argue in Lemma 3.3 about why it is finite.

A divisor is said to be *effective*, if all its coefficients are positive. We obtain a partial order on the divisors by setting $D \leq E :\Leftrightarrow E - D$ is effective. A prime $\mathfrak{p}$ is said to be a *zero* or a *pole* at $a \in F$, if $v_\mathfrak{p}(a)$ is positive, or negative, respectively.

Example 3.1. Let us consider the simplest algebraic function field $k(x)/k$, the *rational function field*. We want to determine the primes of $k(x)$. For this, note that the non-zero prime ideals of $k[x]$ are exactly the ideals generated by a monic irreducible polynomial $p \in k[x]$. The corresponding discrete valuation ring

$$\mathcal{O}_p := \left\{ \tfrac{f}{g} \colon f, g \in k[x], p \nmid g \right\}$$

of $k(x)$ is the localization of $k[x]$ at p and its maximal ideal $\mathfrak{p}$ is a prime of $k(x)$. There is one additional prime, usually denoted by ∞, for which we have to consider the ring $k[1/x]$. Its prime ideal generated by $1/x$ has attached the discrete valuation ring

$$\mathcal{O}_\infty := \left\{ \tfrac{f}{g} \colon f, g \in k[x], \deg f \leq \deg g \right\},$$

as its valuation v_∞ is $-\deg(\cdot)$. This list of primes is exhaustive. Note that the degree of ∞ is one, while the degree of the previously discussed primes is equal to the degree of the corresponding monic irrecducible polynomials.

3.1.1 The Riemann–Roch Theorem

In this section we describe *the* central theorem in the field of algebraic function fields. Its importance is also given by its establishing of an important invariant which can be associated to any function field, namely the genus. Let us first introduce the notions to formulate it.

The Riemann–Roch space of a divisor D is defined as

$$L(D) \coloneqq \{f \in F^\times : (f) + D \geq 0\} \cup \{0\}.$$

It is the k-linear space of all rational functions with pole divisor bounded by D. Denote its dimension as $\ell(D) \coloneqq \dim_k L(D)$. Note that $\ell \colon \operatorname{Div}(F) \to \mathbb{N}$ factors through $\operatorname{Cl}(F)$. This is because if two divisors D and G are linearly equivalent, i. e., $D = G + (f)$ for some $f \in F^\times$, we have that $L(D) \cong L(G)$ by the isomorphism $x \mapsto xf$, implying the equality for $\ell(D)$ and $\ell(G)$ as well.

Theorem 3.1 (Riemann–Roch)**.** *Let F/k be an algebraic function field and $D \in \operatorname{Div}(F)$. Let K be a canonical divisor (for a definiton, see below) and g the genus of F. Then,*

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

Proof. See e. g., [R02, Chapter 6]. For a more constructive proof, see [H02]. □

There are a number of corollaries to the theorem, namely Riemann's inequality, stating that $\ell(D) \geq \deg D - g + 1$, which becomes a equality, if $\deg D \geq 2g + 2$ (unless $\deg D = 2g+2$ *and* D is linearly equivalent to K). Moreover any divisor in the canonical divisor class has degree $2g - 2$ and Riemann–Roch dimension g, which can also be taken as a characterization of the genus. To state how the canonical divisor class looks like, we first introduce so-called Weil differentials.

Weil Differentials. Let F/k be an algebraic function field. By A_F denote the adèle ring of F. It is defined as

$$A_F \coloneqq \Big\{(a_\mathfrak{p}) \in \prod\nolimits_\mathfrak{p} \hat{F}_\mathfrak{p} \colon a_\mathfrak{p} \in \hat{\mathcal{O}}_\mathfrak{p} \text{ for almost all primes } \mathfrak{p} \text{ of } F\Big\},$$

where $\hat{F}_\mathfrak{p}$ and $\hat{\mathcal{O}}_\mathfrak{p}$ are the completions with respect to $\mathfrak{p}$ (see [N92] for a definition). Moreover, for a divisor $D = \sum_\mathfrak{p} n(\mathfrak{p})\mathfrak{p}$, denote by $A_F(D)$ the set of all $(a_\mathfrak{p}) \in A_F$, satisfying $v_\mathfrak{p}(a_\mathfrak{p}) \geq -n(\mathfrak{p})$ for all primes $\mathfrak{p}$. A *Weil differential*, or differential for short, in F is a k-linear function $\omega \colon A_F \to k$, that vanishes on k and on $A_F(D)$ for some divisor D. We denote the F-vector space of differentials on F by Ω. Moreover, for a divisor D we denote the space of differentials that vanish on $A_F(D)$ as $\Omega(D)$, so we have

$$\Omega(D) = \{\omega \in \Omega \colon (\omega) - D \geq 0\} \cup \{0\}.$$

Now, we can attach to any non-zero differential $\omega \in \Omega$ the maximal divisor D, such that ω vanishes on $A_F(D)$. We denote it by (ω). One can show that for $x \in F^\times$ it holds that $(x\omega) = (x) + (\omega)$ and that $\dim_F \Omega = 1$, cf. [R02, Lemma 6.9, Proposition 6.10]. Using this, we can show that exactly all divisors of non-zero differentials lie in the same divisor class in $\operatorname{Cl}(F)$, which we call the *canonical class*. To see this, let $\omega_1, \omega_2 \in \Omega$ be non-zero, then there exists an $x \in F^\times$, such that $\omega_1 = x\omega_2$, by the one-dimensionality of Ω. Moreover, we have $(x\omega_2) = (x) + (\omega_2)$ as stated before, so that (ω_1) and (ω_2) are linearly equivalent. For the other direction, note that if a divisor $D = (x) + (\omega)$ is given with x in $F^\times$ (i.e. linearly equivalent to (ω)), then $D = (x\omega)$ is the divisor of a differential.

3.1.2 Geometric Goppa Codes

In our presentation of geometric Goppa codes, we mostly follow Duursma [D08]. He motivates this class of codes by its good parameters, its multiplicative structure, and efficient code construction and decoding algorithms. Moreover, he mentions secret sharing as an interesting application of Goppa codes. Note that they are a generalization of Reed–Solomon codes, which already have plenty applications in practice.

Linear Codes and Parameters. A *linear code* C of length n over $\mathbb{F}_q$ (q a prime power) is a linear subspace of $\mathbb{F}_q^n$. We denote its dimension as $k := \dim_{\mathbb{F}_q} C$. An important parameter of a code is its minimum distance. For this, we define a metric on the code, which is known as the *Hamming distance*. Let $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}_q^n$, then it is defined as

$$d(\boldsymbol{x}, \boldsymbol{y}) = |\{i \in \{1, \dots, n\} \colon \boldsymbol{x}_i \neq \boldsymbol{y}_i\}|,$$

and counts the number of differing entries of the two vectors. In the following, we exclude the trivial code from our considerations. Then the *minimum distance* is, as its name suggests the smallest possible distance of two elements of the code. We denote it by $d(C)$ or d if no confusion may arise.

Given the parameters n, k, d of a code, we have that $k + d \leq n + 1$, which is known as the Singleton bound. As it is a beneficial property of a code to have large minimum distance, because it allows good error correction, we give codes that attain the Singleton bound, a distinguished name: *maximum distance separable (MDS)*. Geometric Goppa codes over function fields with genus zero are MDS.

Definition 3.1 (Geometric Goppa Codes)**.** Let $F/\mathbb{F}_q$ a global function field, $D = \mathfrak{p}_1 + \cdots + \mathfrak{p}_n$ a divisor, with $\mathfrak{p}_i$, $i = 1, \dots, n$ pairwise distinct primes of F of degree 1, and G another divisor of F with disjoint support. Then the *geometric Goppa code* $C_L(D, G)$ is defined as the image of the linear (evaluation) map

$$\mathrm{ev}_D \colon L(G) \to \mathbb{F}_q^n, \quad f \mapsto (f(\mathfrak{p}_1), \dots, f(\mathfrak{p}_n)).$$

Its kernel is $L(G - D)$, hence we have an isomorphism $L(G)/L(G - D) \cong C_L(D, G)$.

As the divisors D and G need distinct support, the following theorem proves useful.

Theorem 3.2 (Approximation theorem)**.** *For a divisor D and a finite set of primes T, there exists a linearly equivalent divisor that has support outside T.*

Ihara limit. Let $F/\mathbb{F}_q$ be an algebraic function field with exact constant field $\mathbb{F}_q$. Denote its genus by g_F and its number of rational places by $N(F)$. For a family $\mathcal{F} = \{F_i/\mathbb{F}_q\}$ of function fields with exact constant field $\mathbb{F}_q$ and $g_{F_i} \to \infty$ as $i \to \infty$, we define the Ihara limit of $\mathcal{F}$ as

$$A(\mathcal{F}) \coloneqq \limsup_{i\to\infty} \frac{N(F_i)}{g_{F_i}}.$$

The quotient $N(F_i)/g_{F_i}$ is a measure for the capacity of F_i for the purpose of algebraic geometric coding, therefore it is a quest in algebraic coding theory to find families of function fields with large $A(\mathcal{F})$.

Let $N_q(g)$ be the maximal number of primes of degree one for any function field $F/\mathbb{F}_q$ with genus g. It holds that $N_q(g) \in \mathbb{N}$ due to the Serre bound, which implies $N_q(g) \leq q + 1 + g\lfloor 2\sqrt{q}\rfloor$. A proof of this fact can be found in [S93, Chapter V.3]. The *Ihara limit* is then defined as

$$A(q) \coloneqq \limsup_{g\to\infty} \frac{N_q(g)}{g}.$$

In the following section we describe methods to find infinite families of global function fields with a large Ihara limit.

3.2 Infinite Class Field Towers

Galois groups are topological groups and it turns out that they are profinite, i. e., they are isomorphic to a projective limit of discrete finite groups. When studying towers of field extensions, these play a crucial role and deserve a bit of explanation. We tacitly assume all homomorphisms to be continuous.

We start with a definition of projective limits. Let $(G_i)_{i\in I}$ be a family of topological groups and $(I, \leq)$ a filtered index set, i. e., a partially ordered set, such that any two elements have an upper bound in I. Moreover, let $(\varphi_{i,j})_{i\leq j\in I}$ be a family of homomorphisms $\varphi_{i,j}\colon G_j \to G_i$ $(i, j \in I, i \leq j)$, such that $\varphi_{i,i} = \mathrm{id}_{G_i}$ and $\varphi_{i,k} = \varphi_{i,j} \circ \varphi_{j,k}$ for all $i \leq j \leq k \in I$. In this case, we call $((G_i), (\varphi_{i,j}), I)$ a projective system and define its *projective limit* as the topological group $G = \varprojlim_{i\in I} G_i$, together with a family of continuous projections $\pi_i\colon G \to G_i$, satisfying the following universal mapping property: for any topological group H with compatible $\psi_i\colon H \to G_i$ there is a unique continuous homomorphism $f\colon H \to G$ for which the following diagram commutes for all $i \leq j \in I$:

$$\begin{tikzcd}
& H \arrow[d, dotted, "\exists! f"] \arrow[ddl, bend right, "\psi_j"'] \arrow[ddr, bend left, "\psi_i"] & \\
& G \arrow[dl, "\pi_j"'] \arrow[dr, "\pi_i"] & \\
G_j \arrow[rr, "\varphi_{i,j}"'] & & G_i
\end{tikzcd}$$

For any projective system of topological groups, the projective limit exists and is unique up to isomorphism, cf. [K70, Proposition 1.5].

Definition 3.2. A *profinite group* is a topological group which is isomorphic to a projective limit of discrete finite groups. Furthermore, a *finite p-group* is a group of order p^f, $f \in \mathbb{N}$, and a *pro-p-group* is a group isomorphic to a projective limit of discrete finite p-groups.

Profinite groups are exactly the Hausdorff topological groups which are compact and totally disconnected. Furthermore, any Galois group $\mathrm{Gal}(K\,|\,F)$ is a profinite group, and the maximal p-extension of F contained in K, i. e., the composite of all Galois extensions of p-power order contained in K, is a pro-p-group. Pro-p-groups are of interest, as their group cohomology has a very useful interpretation when we talk about class field towers and the cardinality of their Galois groups below.

Therefore, unless stated otherwise, let G be a pro-p-group. A family $(g_i)_{i\in I}$ in G is said to be convergent, if every open subgroup of G contains almost all g_i. We will start with definitions of generator and relation systems of pro-p-groups, as outlined in [NSW13, pp. 224–227].

A *generator system* of G is a convergent family $S = (g_i)$ generating G as a topological group. The *rank* $d(G)$ of G is the cardinality of a minimal generator system of G, where minimal means that it does not contain a proper generating system of G. (By an argument of [NSW13, (3.9.1)] as in Remark 3.1 we will see that this is well-defined for pro-p-groups, although we could define it as the infimum over the cardinalities of all generator systems of G for now.) For a normal subgroup N of G, a *generator system* of N *as a normal subgroup* of G is a convergent family S such that N is the smallest closed normal subgroup of G containing all elements of S.

A pro-p-group A is said to be *free over a set* X, if it is accompanied with a map $i\colon X \to A$, such that every open subgroup contains almost all elements of $i(X)$ and for another map $j\colon X \to H$ with this property into a pro-p-group H, there is a unique homomorphism $\varphi\colon A \to H$ with $j = \varphi \circ i$. For any set X, the free pro-p-group exists and is unique up to unique isomorphism [NSW13, (3.5.14)].

Now consider a presentation of G, given by the exact sequence

$$1 \to R \to A \to G \to 1,$$

where G has a minimal generator system $S = (g_i)_{i\in I}$, A is a free pro-p-group over $X = (x_i)_{i\in I}$ and the map $A \to G$ sends x_i to g_i. The cardinality of a minimal generator system of R as a normal subgroup of A is called the *relation rank* of G and denoted by $r(G)$.

Remark 3.1. We have the following equalities, which relate $d(G)$ and $r(G)$ to the dimensions of certain cohomology groups of G. To avoid a rather longish presentation of cohomology theory, which is not at the center of this thesis, we give a concise description and refer the interested reader to [NSW13, Chapters 1 and 2]: For $n \geq 0$ and a G-module A, let $H^n(G, A)$ denote the n-th cohomology group of G with coefficients in A. The corresponding functor $H^n(G,-)$ from the category of G-modules to the category of abelian groups is defined as the n-th right derivation of the left exact functor $A \mapsto A^G$ (between the same categories). We write $h^n(G) \coloneqq \dim_{\mathbb{F}_p} H^n(G, \mathbb{Z}/p\mathbb{Z})$, where $\mathbb{Z}/p\mathbb{Z}$ is seen with trivial G-action, and obtain $d(G) = h^1(G)$ by [NSW13, (3.9.1)] and $r(G) = h^2(G)$ by [NSW13, (3.9.5)].

Example 3.2. Examples of pro-p-groups are the p-adic integers $\mathbb{Z}_p = \varprojlim_{n\in\mathbb{N}} \mathbb{Z}/p^n\mathbb{Z}$, and of course every finite p-group, where the filtered index set is $\mathbb{N}$ with its usual ordering. As $\mathbb{N}$ is also a filtered set when the divisibility relation is used, an important example of a profinite group is $\hat{\mathbb{Z}} = \varprojlim_{n\in\mathbb{N}} \mathbb{Z}/n\mathbb{Z}$ and we have $\hat{\mathbb{Z}} \cong \prod_p \mathbb{Z}_p$. Note that for both groups $\mathbb{Z}_p$ and $\hat{\mathbb{Z}}$ are topologically generated by a single element, which is why they are also called *pro-cyclic*.

As motivated above, the following theorem by Golod and Šafarevič, gives a criterion on the infinity of a tower of p-extensions. We can think of G as the Galois group of a maximal unramified p-extension of a global field F.

Theorem 3.3 (Golod–Šafarevič)**.** *Let G be a pro-p-group. If G is a finite p-group, then $r(G) > \frac{1}{4}d(G)^2$.*

Proof. See [NSW13, (3.9.7)]. □

For the rest of the chapter, let F be a global field, fix an algebraic closure F^{alg} of F and consider any extension field of F as a subextension of F^{alg}. We give a short summary on important notions on extensions of global fields. Let $K\,|\,F$ be a finite extension, $\mathfrak{p}$ be a prime of F and $\mathfrak{P}$ a prime of K above $\mathfrak{p}$. The *ramification index* of $\mathfrak{P}$ over $\mathfrak{p}$ is the unique integer e satisfying $v_{\mathfrak{P}}(a) = e \cdot v_{\mathfrak{p}}(a)$ for any $a \in F$. We say that $\mathfrak{P}$ is *unramified*, if $e = 1$, and that the extension $K\,|\,F$ is unramified, if any prime of K is. Furthermore, a prime of F is said to be *completely split* in K if it decomposes into $[K : F]$ distinct primes in K. An *abelian* extension is a Galois extension with an abelian Galois group.

Define the *Hilbert class field* of F to be the maximal unramified abelian extension of F. It exists, because composites of unramified extensions are again unramified, and composites of abelian extensions are again abelian. When we set $F = F_0$, we can define a tower of fields extensions, with F_{i+1} being the Hilbert class field of F_i. We obtain the sequence

$$F = F_0 \subset F_1 \subset F_2 \subset \cdots, \qquad \text{with } F_\infty \coloneqq \bigcup F_i,$$

where we would like to know whether it becomes stationary, i. e., whether $F_\infty = F_n$ for some $n \in \mathbb{N}$. If this is not the case, we have an infinite extension. To be able to use the preceding theorem, which only holds for pro-p-groups, we consider the more tractable sequence of p-extensions

$$F = F_0 \subset F_1(p) \subset F_2(p) \subset \cdots, \qquad \text{with } F_\infty(p) \coloneqq \bigcup F_i(p),$$

where $F_{i+1}(p)$ is the maximal unramified abelian p-extension of $F_i(p)$. We can now determine the rank and relation rank of $\mathrm{Gal}(F_\infty(p)\,|\,F)$. (We will see in Lemma 3.1, why this is Galois.) The Golod–Šafarevič inequality then gives a criterion for the infinity of the extension. When Golod and Šafarevič showed their inequality and thus the existence of infinite class field towers over number fields, this was seen as a major achievement, as the contrary would have implied Fermat's last theorem, due to the principal ideal theorem for Hilbert class fields.

However, for our focus on global function fields, this is not yet what we want, as $F_1(p)\,|\,F$ would already be an infinite extension, due to the fact that constant field extensions are unramified, cf. [S93, Theorem III.6.3]. This would result in $\mathbb{F}_q^{\mathrm{alg}}$ as constant field of $F_1(p)$, if $\mathbb{F}_q$ is the constant field of F. To get better control on the construction of codes based on Hilbert class fields, we define class fields with the additional condition on a pre-specified set of primes to be *completely split* in the class field extension. For this, we define the following notions.

Definition 3.3 ([NSW13, p. 452])**.** Let T be a set of primes of a global field F, containing the archimedean (infinite) primes, if F is a number field. We define the *ring of T-integers* of F as $\mathcal{O}_{F,T} := \{a \in F \colon v_{\mathfrak{p}}(a) \geq 0, \text{ for all } \mathfrak{p} \notin T\}$. Moreover, its *$T$-ideal class group* is denoted as $\mathrm{Cl}_T(F)$. It is the quotient $\mathrm{Cl}(F)/\langle T\rangle$, where $\langle T\rangle$ denotes the closed subgroup generated by all primes in T.

For this, we get the following more refined result, using the notation of [NSW13, p. 701]: Let F be a global field, and S, T be arbitrary sets of primes. With $G_S^T(p) := \mathrm{Gal}(F_S^T(p)\,|\,F)$ we denote the Galois group of $F_S^T(p)\,|\,F$, which is the maximal p-extension of F unramified outside S and completely split at every prime of T. We consider the case of $S = \varnothing$ and a non-empty T, where due to Remark 3.1 and class field theory it holds: $d(G_S^T(p)) = d(G_S^T(p)^{\mathrm{ab}}/p)$ and $G_\varnothing^T(p)^{\mathrm{ab}}/p \cong \mathrm{Cl}_T(F)/p$, leading to

$$d(G_\varnothing^T(p)) = d(\mathrm{Cl}_T(F)) = \dim_{\mathbb{F}_p} \mathrm{Cl}_T(F)/p. \tag{3.1}$$

For references on class field theory, see [N92]. Moreover, for a more algorithmic and direct approach to class field theory for global function fields, see [HM13]. With this in mind, we define the *T-Hilbert class field* as the maximal unramified abelian extension, completely split at every prime of T, and analogously, the *(p, T)-Hilbert class field tower*

$$F = F_0 \subset F_1^T(p) \subset F_2^T(p) \subset \cdots, \qquad \text{with } F_\infty^T(p) = \bigcup F_i^T(p),$$

where $F_n^T(p)$ is the maximal abelian unramified p-extension of F_{n-1}^T completely split at every prime of T_n, with $T = T_1$ and T_n is the set of primes of F_{n-1}^T above T. In the following lemma, we show some basic facts about these notions.

Lemma 3.1. *Let T be a finite set of primes, p a prime number and F_n^T, $F_n^T(p)$, $F_\infty^T(p)$ and $F_\varnothing^T(p)$ be defined as above. Then it holds:*

1. *$F_n^T(p) \subseteq F_n^T$, for $n \in \mathbb{N}$. Hence, if F has an infinite (p, T)-Hilbert class field tower, than it also has an infinite T-Hilbert class field tower.*

2. *$F_n^T\,|\,F$ and $F_n^T(p)\,|\,F$ are Galois, for $n \in \mathbb{N}$.*

3. *F_∞ is the maximal pro-solvable (i. e., its Galois group is isomorphic to a projective limit of solvable groups) unramified extension of F.*

4. *$F_\infty^T(p) = F_\varnothing^T(p)$, i. e., $F_\infty^T(p)$ is the maximal unramified p-extension, completely split at every prime of T.*

5. *If T is non-empty and F a global function field, with exact constant field $\mathbb{F}_q$, then the exact constant field of the T-Hilbert class field is $\mathbb{F}_{q^d}$, where d is the greatest common divisor of the degrees of primes in T.*

Proof. 1. We use induction, as e. g., in [T10, Lemma 1]. $F_1^T(p) \subseteq F_1$ holds trivially as any p-extension is also a field extension of F. Therefore, assume $F_n^T(p) \subseteq F_n^T$ for an $n \in \mathbb{N}$ and consider the following diagram of field extensions:

$$\begin{tikzcd}
& F_{n+1}^T F_n^T F_{n+1}^T(p) \arrow[dl, dash] \arrow[dd, dash] & \\
F_{n+1}^T \arrow[dd, dash] & & \\
& F_n^T F_{n+1}^T(p) \arrow[dl, dash] \arrow[dr, dash] & \\
F_n^T \arrow[dr, dash] & & F_{n+1}^T(p) \arrow[dl, dash] \\
& F_n^T(p) &
\end{tikzcd}$$

Note that composites of unramified extensions are again unramified, and composites of abelian extensions are again abelian. Therefore, $F_{n+1}^T F_n^T F_{n+1}^T(p) \,|\, F_n^T$ is an unramified abelian extension. This implies $F_{n+1}^T(p) \subseteq F_{n+1}^T$, due to maximality.

2. We use induction, as e. g., in the proof of [NX01, Theorem 2.7.7]. Separability can be observed directly. Assume that F_n^T is normal for an $n \in \mathbb{N}$. For an embedding $\sigma\colon F_{n+1}^T \to F^{\mathrm{alg}}$ over F, we have that $\sigma(F_n^T) = F_n^T$. Therefore, also $\sigma(F_{n+1}^T) \,|\, \sigma(F_n^T) = \sigma(F_{n+1}^T) \,|\, F_n^T$ is abelian unramified. By the maximality, we have that $\sigma(F_{n+1}^T) \subseteq F_{n+1}^T$, and hence, $F_{n+1}^T \,|\, F$ is normal. The same argument applies to $F_n^T(p)$.

3. Let L denote the maximal pro-solvable unramified extension of F and note that $F_\infty \subseteq L$. For the other inclusion, take $x \in L$. Without loss of generality we can assume $F(x) \,|\, F$ to be Galois. Note that $F(x) \,|\, F$ is unramified and its Galois group G is solvable, as G is a finite quotient of a pro-solvable group. Therefore, by Galois theory it follows that there exist

$$F = F^{(0)} \subsetneq F^{(1)} \subsetneq \cdots \subsetneq F^{(m)} = F(x),$$

such that $F^{(i+1)} \,|\, F^{(i)}$ is abelian unramified. Using induction we can show that $F^{(i)} \subseteq F_i$, and from that $L \subseteq F_\infty$ follows.

4. Note that we have $F_n^T(p) \subseteq F_\varnothing^T(p)$ for any $n \in \mathbb{N}$, and hence, $F_\infty^T(p) \subseteq F_\varnothing^T(p)$. The other direction is a corollary of the previous item, as the Galois groups are pro-p, and p-groups are solvable.

5. See [NX01, Proposition 2.5.7]. □

Concerning the last statement, if T is non-empty and the greatest common divisor of the degrees of the primes in T is one, we have an extension which leaves the exact constant field unchanged. We call extensions with this property *geometric extensions*.

The following theorem gives a criterion of the infinity of (p, T)-Hilbert class field towers. Here, μ_p denotes the p-th roots of unity.

Theorem 3.4 (cf. [NSW13, (10.10.5)])**.** *Let F be a global field and T a finite set of primes of F. Let r be the number of archimedean primes of F, which is zero in the function field case, $p \neq \operatorname{char} F$ be a prime number and*

$$F = F_0 \subset F_1^T(p) \subset F_2^T(p) \subset \cdots, \qquad \textit{with } F_\infty^T(p) = \bigcup F_i^T(p),$$

the (p, T)-Hilbert class field tower of F. If

$$\dim_{\mathbb{F}_p} \mathrm{Cl}_T(F)/p \geq 2 + 2\sqrt{r + \delta_p + |T|},$$

where $\delta_p = 1$, if $\mu_p \subseteq F$ and 0 otherwise, then $F_\infty^T(p)\,|\,F$ is infinite.

Proof. Assume that $G = \mathrm{Gal}(F_\infty^T(p)\,|\,F)$ is finite. Starting from the Golod–Šafarevič inequality $\frac{1}{4}d(G)^2 < r(G)$, we deduce by elementary term manipulations:

$$(d(G) - 2)^2 < 4(r(G) - d(G) + 1). \tag{$*$}$$

We use the following result of [NSW13, (10.7.12)] as a blackbox:

$$r(G) - d(G) + 1 \leq r + \delta_p + |T|.$$

With this, $(*)$ further simplifies to

$$d(G) < 2 + 2\sqrt{r + \delta_p + |T|}.$$

From class field theory we have noted above in (3.1), that

$$d(G) = \dim_{\mathbb{F}_p} \mathrm{Cl}_T(F)/p,$$

leading to $\dim_{\mathbb{F}_p} \mathrm{Cl}_T(F)/p < 2 + 2\sqrt{r + \delta_p + |T|}$, which is a contradiction to our initial assumption. □

As we are mainly interested in the case of global function fields, we derive the following corollary.

Corollary 3.1 (cf. [NX01, Theorem 2.7.7])**.** *Let $F/\mathbb{F}_q$ be a global function field with exact constant field $\mathbb{F}_q$ and T a finite set of primes of F. Let $p \neq \operatorname{char} \mathbb{F}_q$ be a prime number and*

$$F = F_0 \subset F_1^T(p) \subset F_2^T(p) \subset \cdots, \qquad \textit{with } F_\infty^T(p) = \bigcup F_i^T(p),$$

the (p,T)-Hilbert class field tower of F. If

$$\dim_{\mathbb{F}_p} \mathrm{Cl}_T(F)/p \geq 2 + 2\sqrt{|T| + \delta_p(q)},$$

where $\delta_p(q) = 1$, if $p \mid (q-1)$, and 0 otherwise, then $F_\infty^T(p)\,|\,F$ is infinite.

Proof. This is an immediate consequence of Theorem 3.4 and the fact that $\mu_p \subseteq \mathbb{F}_q$ if and only if $p \mid (q-1)$. □

Proposition 3.1 (cf. [NX01, Theorem 2.7.6])**.** *Let $F/\mathbb{F}_q$ be a global function field with exact constant field $\mathbb{F}_q$, $g_F > 1$ and T a finite set of primes* of degree one *of F. For the family $\mathcal{F} = \{F_i^T(p)/\mathbb{F}_q\}$ of algebraic function fields corresponding to the (T,p)-Hilbert class field tower over F as in Corollary 3.1, the Ihara limit $A(\mathcal{F})$ exists and satisfies*

$$A(\mathcal{F}) \geq \frac{|T|}{g_F - 1}.$$

Proof. Let $F_i := F_i^T(p)$. We first need to determine the genus g_{F_i} and the number of primes of degree 1 of F_i. We have that since $F_i\,|\,F$ is a finite unramified, separable, and geometric extension, $2g_{F_i} - 2 = [F_i : F] \cdot (2g_F - 2)$, due to the Riemann–Hurwitz genus formula (see e. g., [NX01, Theorem 1.3.10]). Furthermore, let N_i denote the number of primes of degree 1 of F_i. Then $N_i \geq |T_i| = [F_i : F] \cdot |T|$, by definition of T. This implies that

$$A(\mathcal{F}) := \limsup_{i\to\infty} \frac{N_i}{g_{F_i}} \geq \lim_{i\to\infty} \frac{[F_i : F] \cdot |T|}{[F_i : F](g_F - 1) + 1} = \frac{|T|}{g_F - 1}.$$ □

Using this, we can now construct a family of function fields with an Ihara limit that is large enough for our purposes using Kummer and Artin–Schreier extensions. Before we start, let us quickly remind us of the theory of both types of extensions for global function fields in the following lemma.

Lemma 3.2 (Kummer and Artin–Schreier extensions)**.** *Let $F/\mathbb{F}_q$ a global function field with exact constant field $\mathbb{F}_q$, with $\operatorname{char} \mathbb{F}_q = p > 0$.*

1. *Let $n \in \mathbb{N}_{\geq 2}$ so that $\mu_n \subseteq \mathbb{F}_q$ (which by convention implies $\gcd(p,n) = 1$) and suppose there is an $u \in F$, such that $u \neq w^d$ for all $w \in F$, $d \mid n$, $d > 1$. Then $F(y)$ with $y^n = u$ is called a* Kummer extension *of F. We have:*
   - *a) $F(y)\,|\,F$ is a cyclic Galois extension of degree n.*
   - *b) Let $\mathfrak{p}$ be a prime of F and $\mathfrak{P}$ a prime of $F(y)$ above $\mathfrak{p}$. Then its ramification index is $e = n/\gcd(n, v_{\mathfrak{p}}(u))$, where* gcd *denotes the greatest common divisor.*

c) *Let $\mathbb{F}_{q^k}$, $k \in \mathbb{N}_{\geq 1}$ denote the exact constant field of $F(y)$, then*

$$g_{F(y)} = 1 + \frac{n}{k}\left(g_F - 1 + \frac{1}{2} \sum_{\mathfrak{p} \in \mathbb{P}(F)} \left(1 - \frac{\gcd(n, v_{\mathfrak{p}}(u))}{n} \right) \deg \mathfrak{p} \right).$$

2. *Suppose there is an $u \in F$, such that $u \neq w^p - w$ for all $w \in F$. Then $F(y)$ with $y^p - y = u$ is called an* Artin–Schreier extension *of F. We have:*
   a) *$F(y)\,|\,F$ is a cyclic Galois extension of degree p.*
   b) *A prime $\mathfrak{p}$ of F is unramified in $F(y)\,|\,F$, if and only if there is a $z \in F$, such that $v_{\mathfrak{p}}(u - (z^p - z)) \geq 0$.*

Proof. See [S93, Propositions III.7.3 and III.7.8]. □

Using this, we can now construct the family of function fields as needed.

Theorem 3.5 ([CCX12b, Theorem 2.6])**.** *For every $q \geq 8$, except perhaps for $q = 11$ or 13, there exists a family of function fields over $\mathbb{F}_q$ such that the Ihara limit $A(\mathcal{F})$ exists and it satisfies $A(\mathcal{F}) > 1 + \frac{1+\delta_2(q)}{\log_2 q}$, where $\delta_p(q) = 1$, if $p \mid (q-1)$ and 0 otherwise.*

Proof. Using Proposition 3.1, we want to find a global function field $F/\mathbb{F}_q$ with exact constant field $\mathbb{F}_q$, $g_F > 1$, and an unramified abelian extension $K\,|\,F$ of degree 2, such that enough primes are completely split in K. We first assume that $q \geq 17$.

1. Let additionally q be odd. Then $\mu_2 \subseteq \mathbb{F}_q$ and we can construct Kummer extensions of degree 2. As $x \mapsto x^2$ is a group endomorphism of $\mathbb{F}_q^\times$, there are $\frac{q-1}{2}$ square elements in $\mathbb{F}_q$, for odd q. So, choose $t_1, \ldots, t_7 \in (\mathbb{F}_q)^2$ non-zero and consider for each $i = 1, \ldots, 7$ the extension $K_i = \mathbb{F}_q(x, y_i)$, where $y_i^2 = x + t_i$. We have that x is completely split in K_i, $i = 1, \ldots, 7$, as $x = y_i^2 - t_i = (y_i + \sqrt{t_i})(y_i - \sqrt{t_i})$, and $\pm\sqrt{t_i} \in \mathbb{F}_q$.

   Now let $F = \mathbb{F}_q(x, y)$, with $y^2 = \prod_{i=1}^{7}(x + t_i)$. We have that $F \subseteq K := K_1 \cdots K_7$, as $y^2 = (y_1 \cdots y_7)^2$ and $[F : \mathbb{F}_q(x)] = 2$. We want to show that $K\,|\,F$ is an unramified abelian extension. For this note that each $K_i\,|\,\mathbb{F}_q(x)$ ramifies at exactly one finite prime with ramification index 2, while $F\,|\,\mathbb{F}_q(x)$ ramifies at all these primes with the same index. Therefore $K\,|\,F$ is unramified, as the ramification index is multiplicative and all ramification of $K\,|\,\mathbb{F}_q(x)$ happens in $F\,|\,\mathbb{F}_q(x)$ already.

   Note that $\mathrm{Gal}(K_i\,|\,\mathbb{F}_q(x)) \cong \mathbb{Z}/2\mathbb{Z}$, and that we have an injective map

   $$\mathrm{Gal}(K\,|\,\mathbb{F}_q(x)) \hookrightarrow \prod_i \mathrm{Gal}(K_i\,|\,\mathbb{F}_q(x)), \quad \sigma \mapsto \left(\sigma|_{K_i}\right)_i.$$

   A calculation using the fact that there is a bijection of the subgroups A of $\mathbb{F}_q^\times/\mathbb{F}_q^{\times 2}$ and the abelian extensions obtained by the adjunction of the roots of A to $\mathbb{F}_q$, then yields that $\mathrm{Gal}(K\,|\,\mathbb{F}_q(x)) \cong (\mathbb{Z}/2\mathbb{Z})^7$. From this, due to the multiplicativity of the extension degrees, we have that $\mathrm{Gal}(K\,|\,F) \cong (\mathbb{Z}/2\mathbb{Z})^6$. Moreover, note that ∞ and the two primes above x are completely split in $K\,|\,F$, so collect them in a set T and note that by (3.1) we have $\dim_{\mathbb{F}_2} \mathrm{Cl}_T(F)/2 = 6$, which is equal to $2 + 2\sqrt{|T|+1}$. As $g_F = 3$ by item 1c of Lemma 3.2, we can use Proposition 3.1 to obtain an (T, p)-Hilbert class field tower over F. For its family $\mathcal{F} = \{F_i^T(p)/\mathbb{F}_q\}$ of algebraic function fields, it holds that $A(\mathcal{F}) \geq \frac{|T|}{g_F - 1} = \frac{3}{2}$, which is larger than $1 + 2/\log_2 q$.

2. Now let q be even. Then $\mathrm{char}\,\mathbb{F}_q = 2$ and we can construct Artin–Schreier extensions of degree 2, which is analogous to the Kummer extension case.

3. Let us consider the remaining cases $q \in \{8, 9, 16\}$. For $q = 8$, we have by the by the Drinfeld–Vlăduţ bound for the Ihara limit of square-powers q, which is

$$A(q) = \sqrt{q} - 1,$$

cf. [I82; VD83], that there is a family $\mathcal{F}$ of global function fields over $\mathbb{F}_8$ with $A(\mathcal{F}) = 2\sqrt{2} - 1 \geq \frac{3}{2} \geq 1 + \frac{1}{\log_2(8)}$.

For $q = 9$ and $q = 16$ we get by the following bound on the Ihara limit, due to [Z85; BGS05], that there is a family $\mathcal{F}$ over $\mathbb{F}_9$ and $\mathbb{F}_{16}$, such that $A(\mathcal{F}) = 2$ and $A(\mathcal{F}) = 3$, respectively:

$$A(q^3) = \frac{2(q^2 - 1)}{q + 2}.$$

This leads to $A(\mathcal{F}) > 1 + \frac{1+\delta_2(q)}{\log_2(q)}$ in both cases. □

3.3 Riemann–Roch System of Equations

In this section we introduce equation systems on the dimension of Riemann–Roch spaces of a special form. For this, let F be an algebraic function field with exact constant field $\mathbb{F}_q$, $s \in \mathbb{N}_{\geq 1}$, $Y_i \in \mathrm{Cl}(F)$ and $m_i \in \mathbb{Z} \setminus \{0\}$, for $i = 1, \ldots, s$. We call a system

$$\{\ell(m_i X + Y_i) = 0\}_{i=1}^s$$

a *Riemann–Roch system of equations* in X. A solution for the system is a $[G] \in \mathrm{Cl}(F)$, satisfying all equations when substituted for X, cf. [CCX12b, Definiton 3.1]. While it is easy to find a solution by choosing $[G]$ with a degree such that $\deg(m_i X + Y_i) < 0$, we aim for solutions where the degree of $[G]$ is pre-specified.

Before we give a criterion on the solvability of these systems, let us argue about the finiteness of J_F in the following lemma.

Lemma 3.3 ([R02, Lemma 5.5 and 5.6])**.** *Let F be a global function field with exact constant field $\mathbb{F}_q$. The number of effective divisors E_d of degree $d \in \mathbb{N}$ is finite. Moreover, the cardinality $h_F := |J_F|$ of the zero divisor class group J_F is finite.*

Proof. For the first claim, let $x \in F$ be transcendental over $\mathbb{F}_q$ and consider the finite primes of $\mathbb{F}_q(x)$. As in Example 3.1, they are given by the monic irreducible polynomials of $\mathbb{F}_q[x]$, of which there are only finitely many of a fixed degree $d \in \mathbb{N}$. Note that $F|\mathbb{F}_q(x)$ is a finite extension and thus, there are only finitely many primes of F of a given degree. For an effective divisor of degree d, the degrees of primes in its support are bounded by d as well, and hence are finitely many. As the bound holds also for the coefficients, the number of effective divisors of a given degree is finite.

For the second claim, we want to show that we can find for any divisor A of degree 0, an effective divisor of a specified degree $s \in \mathbb{N}$. For this, let D be a divisor of degree 1. By the theorem of Riemann–Roch (Theorem 3.1) it holds that $\ell(sD + A) \geq 1$. So we find an $f \in L(sD + A)$ and can define $B := (f) + sD + A$, which is effective, of degree s and linearly equivalent to $sD + A$. With this, we have that $h_F \leq E_s$, which is finite by the first claim. □

h_F is called the *class number* of F. We have the following theorem about solutions for Riemann–Roch systems of equations. Here, $J_F[m_i]$ denotes the m_i-torsion in the zero divisor class group, i. e., $J_F[m_i] = \{[D] \in J_F \colon m_i[D] = 0\}$.

Theorem 3.6 ([CCX12b, Theorem 3.2]). *Consider the Riemann–Roch system of equations*

$$\{\ell(m_i X + Y_i) = 0\}_{i=1}^{s}.$$

Let $d_i = \deg Y_i$, for $i = 1, \ldots, s$. Denote by E_r the number of effective divisors of degree r in $\mathrm{Div}(F)$ *for $r \geq 0$, and 0 for $r < 0$. Let $d \in \mathbb{Z}$ and define $r_i = m_i d + d_i$ for $i = 1, \ldots, s$. If*

$$h_F > \sum_{i=1}^{s} E_{r_i} \cdot |J_F[m_i]|,$$

then the Riemann–Roch system of equations has a solution $[G] \in \mathrm{Cl}^{(d)}(F)$.

Proof. We look at the equations with non-negative degree only, and so consider for $i \in S := \{1 \leq i \leq s \colon r_i \geq 0\}$ the maps

$$\begin{aligned} \varphi_i &\colon \mathrm{Cl}^{(d)}(F) \to \mathrm{Cl}^{(m_i \cdot d)}(F), \quad && X \mapsto m_i X, \\ \psi_i &\colon \mathrm{Cl}^{(m_i \cdot d)}(F) \to \mathrm{Cl}^{(r_i)}(F), \quad && X \mapsto X + Y_i. \end{aligned}$$

Note that each $X' \in \operatorname{im} \varphi_i$ has exactly $|J_F[m_i]|$ preimages, as the kernel of φ_i is $J_F[m_i]$ by definition. Furthermore, ψ_i is injective, as $X \mapsto X - Y_i$ is its inverse.

We set $\rho_i := \psi_i \circ \varphi_i$ and obtain $|\rho_i^{-1}(Z)| \leq |J_F[m_i]|$ for any effective divisor class $Z \in \mathrm{Cl}^{(r_i,+)}(F)$. From this, we obtain $|\rho_i^{-1}(\mathrm{Cl}^{(r_i,+)}(F))| \leq |\mathrm{Cl}^{(r_i,+)}(F)| \cdot |J_F[m_i]| \leq E_{r_i} \cdot |J_F[m_i]|$, where E_{r_i} is the number of effective divisors of degree r_i. Hence,

$$|\bigcup_{i \in S} \rho_i^{-1}(\mathrm{Cl}^{(r_i,+)}(F))| \leq \sum_{i \in S} E_{r_i} \cdot |J_F[m_i]| =: z.$$

Since $h_F = |\mathrm{Cl}^{(d)}(F)| > z$ by assumption, there is an $[G] \in \mathrm{Cl}^{(d)}(F)$ which is not in $\bigcup_{i \in S} \rho_i^{-1}(\mathrm{Cl}^{(r_i,+)}(F))$. For this, it holds that $\rho_i([G]) \in \mathrm{Cl}^{(r_i)}(F) \setminus \mathrm{Cl}^{(r_i,+)}(F)$.

Hence, there are no effective representatives of $\rho_i([G])$ and thus no $f \in F^\times$, such that $[(f)] + \rho_i([G]) \geq 0$. It follows that $\ell(\rho_i([G])) = 0$ for $i \in S$ and finally, that $[G]$ is a solution of the system. □

3.4 Torsion Limit

Let $r > 1$ and denote by $J_F[r]$ the r-torsion in the zero divisor class group.

Definition 3.4 (Torsion limit, [CCX12b, Definition 2.2])**.** For a family $\mathcal{F} = \{F_i/\mathbb{F}_q\}$ of function fields with $g_{F_i} \to \infty$ as $i \to \infty$, we define the *r-torsion limit* of $\mathcal{F}$ as

$$J_r(\mathcal{F}) := \liminf_{i\to\infty} \frac{\log_q |J_{F_i}[r]|}{g_{F_i}}.$$

Moreover, we want to define a notion of the least possible r-torsion of a family of function fields for which the Ihara limit exceeds a given value A. Define,

$$J_r(q, A) := \liminf_{\mathcal{F}\in\mathfrak{F}(A)} J_r(\mathcal{F}),$$

where $\mathfrak{F}(A)$ is a set of families $\mathcal{F}$ of function fields over $\mathbb{F}_q$, such that the genus tends to ∞ and its Ihara limit is greater or equal to A.

Theorem 3.7. *Let $\mathbb{F}_q$ be a finite field and let $p > 1$ be a prime.*

1. *$J_p(q, A(q)) \leq \frac{1+\delta_p(q)}{\log_p q}$, where $\delta_p(q) = 1$, if $p \mid (q-1)$ and 0 otherwise.*
2. *If q is square and $p \mid q$, then $J_p(q, A(q)) \leq \frac{1}{(\sqrt{q}+1)\cdot\log_p q}$.*

Proof. See [CCX12b, Theorem 2.3]. □

Corollary 3.2. *The family of function fields of Theorem 3.5 satisfies $A(\mathcal{F}) > 1 + J_2(\mathcal{F})$.*

Proof. By Theorem 3.7, we have for $p = 2$

$$A(\mathcal{F}) > 1 + \frac{1+\delta_p(q)}{\log_p q} \geq 1 + J_p(q, A(q)).$$

As $J_p(q,\cdot)$ is monotone by definition, it holds $J_p(q, A(q)) \geq J_p(q, A(\mathcal{F}))$. □

3.5 Arithmetic Secret Sharing Schemes

In this section we bridge the gap to the previous chapter and the provable-security notions of secret sharing. In the following, let $\mathcal{M}$ be a finite $\mathbb{F}_q$-algebra, which is finitely-generated as an $\mathbb{F}_q$-module, and n, $d \in \mathbb{N}_{\geq 1}$. For an $\boldsymbol{x} \in \mathcal{M} \times \mathbb{F}_q^n$, and a set $\varnothing \neq A \subseteq \{1, \ldots, n\}$, denote by $\pi_0 \colon \mathcal{M} \times \mathbb{F}_q^n \to \mathcal{M}$ the projection of $\boldsymbol{x}$ to its $\mathcal{M}$-component, and $\pi_A \colon \mathcal{M} \times \mathbb{F}_q^n \to \mathbb{F}_q^{|A|}$ the projection of $\boldsymbol{x}$ to the coordinates of A.

Definition 3.5 (Codex, cf. [CCX12a; J13])**.** Let C be a proper $\mathbb{F}_q$-linear subspace of $\mathcal{M} \times \mathbb{F}_q^n$ and $\Gamma = (\mathcal{A}, \mathcal{B})$ an access structure on a player set P. We say that C is an *(n, d, Γ)-codex for $\mathcal{M}$ over $\mathbb{F}_q$*, if the following holds:

1. $\pi_0(C) = \mathcal{M}$,
2. There is $\mathcal{A}$-reconstruction of d-fold products, i. e., for any $A \in \mathcal{A}$ there is a linear (reconstruction) map $\rho^A \colon \mathbb{F}_q^n \to \mathcal{M}$, such that
   a) $\rho^A(\prod_{i=1}^d c_i) = \prod_{i=1}^d \pi_0(c_i)$, for all $c_1, \dots, c_d \in C$, and
   b) $\pi_0(c) = \rho^A \circ \pi_A(c)$ for all $c \in C$.
3. C has $\mathcal{B}$-privacy, i. e., for any $B \in \mathcal{B}$, $B \neq \varnothing$, the projection $\pi_{\{0\} \cup B} \colon C \to \mathcal{M} \times \pi_B(C)$ is surjective. If additionally $\pi_B(C) = \mathbb{F}_q^{|B|}$, then C is said to have *uniformity*.

For the following definition, recall the definitions from the beginning of Section 1.1.

Definition 3.6 (Arithmetic secret sharing scheme)**.** Let $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$ be an n-player distribution scheme with message space $\mathcal{M}$ and share spaces $\mathbb{F}_q$, and $\Gamma = (\mathcal{A}, \mathcal{B})$ an access structure on a player set P. Let $d \geq 2$ and Sh, Rec be defined via an (n, d, Γ)-codex C for $\mathcal{M}$ over $\mathbb{F}_q$ as

$$\mathsf{Sh}(m) = \pi_P(c), \text{ for } c \leftarrow \pi_0^{-1}(m), m \in \mathcal{M},$$
$$\mathsf{Rec}(\boldsymbol{x}, j) = \begin{cases} \perp, & \text{if } \psi(\boldsymbol{x}, j) = \perp, \\ \rho^{\psi(\boldsymbol{x},j)} \circ \varphi(\boldsymbol{x}), & \text{otherwise,} \end{cases} \quad \text{for } \boldsymbol{x} \in (\mathbb{F}_q \cup \{\lozenge\})^n, j \in \{0, \dots, n\},$$

where $\varphi \colon (\mathbb{F}_q \cup \{\lozenge\})^n \to \mathbb{F}_q^n$ is the identity map on $\mathbb{F}_q^n$ and sends the empty share sign $\lozenge$ to 0, $\psi \colon (\mathbb{F}_q \cup \{\lozenge\})^n \times \{0, \dots, n\} \to \mathcal{A} \cup \{\perp\}$ selects a set $A \in \mathcal{A}$ containing player P_j, if $j \neq 0$ and not containing players of $\lozenge$-entries, or attains the value $\perp$ if there is no such set; and ρ^A is the reconstruction map guaranteed by Definition 3.5.

If $\mathcal{B} \neq \varnothing$ we call Σ a *(n, d, Γ)-arithmetic secret sharing scheme.* It is said to have *uniformity* if C is uniform. In the case of $\mathcal{B} = \varnothing$, Σ is called a *$(n, d, \mathcal{A})$-arithmetic information dispersal algorithm (IDA).* If $\Gamma = \Gamma(t, r)$ with $t \geq 1$, we may call Σ an (n, t, d, r)-arithmetic secret sharing scheme for short.

To justify the name we have the following lemma. For this, recall the definitions from the beginning of Section 1.2, including Definition 1.6.

Lemma 3.4. *Let Σ be an (n, d, Γ)-arithmetic secret sharing scheme for $\mathcal{M}$ over $\mathbb{F}_q$. Then Σ is an n-player perfect-privacy linear $\mathcal{C}_{t \to l}$-arithmetic SSS over $\mathbb{F}_q$ with access structure Γ, message space $\mathcal{M}$ and share spaces $\mathbb{F}_q$. Moreover, its* Sh *function is linear and is compatible with d-fold multiplications as in Remark 1.3.*

Proof. 1. We first show that the advantage of any $\mathcal{B}$-privacy adversary A in Experiment 1.1 is zero. This argument follows [CCX12a, p. 3] and [C+09, Theorem 1]. Let $B \in \mathcal{B}$, $B \neq \varnothing$, and C the codex of the arithmetic secret sharing scheme Σ.

Then the privacy of the codex implies that $\pi_{\{0\}\cup B}\colon C \to \mathcal{M} \times \pi_B(C)$ sending $c \in C$ to $(\pi_0(c), \pi_B(c))$, has the property that for each $m \in \mathcal{M}$, $\boldsymbol{s} \in \pi_B(C)$ there is a $c \in C$ with $\pi_{\{0\}\cup B}(c) = (m, \boldsymbol{s})$. Moreover, note that as C is finite, their number is finite and equal to the cardinality of the kernel of $\pi_{\{0\}\cup B}$ and therefore independent of the choice of $(m, \boldsymbol{s})$.

Hence, if $c \leftarrow C$ uniformly at random, we have that $\pi_{\{0\}\cup B}(c)$ has the uniform distribution on $\mathcal{M} \times \pi_B(C)$ and hence, also that $\pi_0(c)$ has the uniform distribution on $\mathcal{M}$. Furthermore, $\pi_0(c)$ and $\pi_B(c)$ are independently distributed, i. e., knowing the outcome of $\pi_B(c)$ does not help distinguishing the uniformly distributed elements of $\pi_0(C)$, as $c \leftarrow \pi_0^{-1}(m)$, for $m \in \mathcal{M}$ by definition of Sh.

2. Our next step is to show that the advantage of any $\mathcal{A}$-erasure reconstruction adversary B in Experiment 1.2 is zero. For this note that by definition B has to leave at least one set $A \in \mathcal{A}$ uncorrupted, so ψ in the definition of Rec will never output $\perp$ in this case. The guaranteed existence of the linear reconstruction map ρ^A for an uncorrupted set $A \in \mathcal{A}$ then shows that the result is $m = \pi_0(c)$, $c \in C$, even if c has been projected to its A-component.

3. Linearity can be seen directly as $\mathcal{M}$, $\mathbb{F}_q^n$ are finitely-generated $\mathbb{F}_q$-modules and Sh is a linear map.

4. The compatibility of d-fold multiplications is derived directly from the fact that the codex has $\mathcal{A}$-reconstruction of d-fold products.

5. Σ is $\mathcal{C}_{t\to l}$-arithmetic. For this, note that the security against $\mathcal{C}_{t\to l}$-arithmetic $\mathcal{B}$-privacy adversaries follows directly from Lemma 1.1 as Sh is linear. The property for the reconstruction follows from the previous item and Lemma 1.2. □

3.5.1 Construction of Arithmetic SSS

In this section we set as message space $\mathcal{M} := \mathbb{F}_q^k$ and aim to construct an infinite family of codices, which can be interpreted as arithmetic secret sharing schemes as described above.

Theorem 3.8 (cf. [CCX12b, Theorem 4.11], [CCX12a, Theorem 6])**.** *Let $d \geq 2$, $(\omega) \in \mathrm{Div}(F)$ a canonical divisor, $\mathfrak{q}_1, \ldots, \mathfrak{q}_k$, $\mathfrak{p}_1, \ldots, \mathfrak{p}_n$ be primes of degree one of $F/\mathbb{F}_q$. We write $Q := \sum \mathfrak{q}_i$ and $P_I := \sum_{i\in I} \mathfrak{p}_i$ for an index set $I \subseteq \{1, \ldots, n\}$. If the system*

$$\{\ell(dX - P_A) = 0, \ell((\omega) - X + P_B + Q) = 0\}_{A \in \min \mathcal{A}, B \in \max \mathcal{B}}$$

is solvable, then there is a solution $G \in \mathrm{Div}(F)$ such that $C_L(G, D)$ is an (n, d, Γ)-codex for $\mathbb{F}_q^k$ over $\mathbb{F}_q$ with uniformity. (Here $\min \mathcal{A}$ denotes the min-terms of $\mathcal{A}$ and $\max \mathcal{B}$ the max-terms of $\mathcal{B}$.)

Proof. For the geometric Goppa code $C := C_L(G, D)$ we need that G and D have disjoint support. If this is not the case, we can simply use the approximation theorem

(Theorem 3.2) to obtain a solution $G \in \mathrm{Div}(F)$ satisfying the equation system and having a support disjoint from D.

As G fulfills the conditions $\ell(dG - P_A) = 0$ for $A \in \mathcal{A}$, we want to show that this implies the $\mathcal{A}$-reconstruction property of d-fold products for C. We first observe that for $f_1, \ldots, f_d \in L(G)$, it holds that $f_1 \cdots f_d \in L(dG)$. Hence, let $f \in L(dG)$ and assume that $f(P_A) = 0$. Then $f \in L(dG - P_A)$, which is $\{0\}$, as $\ell(dG - P_A) = 0$ by assumption. Therefore, $f = 0$ and trivially $f(Q) = 0$. Hence, in the terminology used above, $\ker(\pi_A) \cap C \subseteq \ker(\pi_0)$, which implies that for any two $c, c' \in C$, with $\pi_A(c) = \pi_A(c')$, we have that $\pi_0(c) = \pi_0(c')$. From this, we can construct the linear reconstruction function ρ^A with $\pi_0(x) = \rho^A \circ \pi_A(x)$, for all $x \in C$.

Moreover, we have that G satisfies $\ell((\omega) - G + P_B + Q) = 0$ for $B \in \mathcal{B}$. Let $B = \{i_1, \ldots, i_t\}$. In the following we show that this implies $\mathcal{B}$-privacy of C. For this, note that the kernel of the evaluation map $\mathrm{ev}\colon L(G) \to \mathbb{F}_q^{k+t}$, given by

$$f \mapsto (f(\mathfrak{q}_1), \ldots, f(\mathfrak{q}_k), f(\mathfrak{p}_{i_1}), \ldots, f(\mathfrak{p}_{i_t}))$$

is $L(G - Q - P_B)$. This implies that the dimension of the image space of ev is $\ell(G) - \ell(G - Q - P_B)$. Using the Riemann–Roch theorem (Theorem 3.1), we obtain for a canonical divisor (ω):

$$\begin{aligned} \ell(G) &= \ell((\omega) - G) + \deg(G) - g + 1, \\ \ell(G - Q - P_B) &= \ell((\omega) - G + Q + P_B) + \deg(G - Q - P_B) - g + 1, \end{aligned}$$

and

$$\ell(G) - \ell(G - Q - P_B) = \ell((\omega) - G) - \ell((\omega) - G + Q + P_B) + \deg(Q + P_B).$$

As $Q + P_B$ is effective, we have that $\ell((\omega) - G) \leq \ell((\omega) - G + Q + P_B)$, which is zero by assumption. Hence, the dimension of the image space of ev is $\deg(Q + P_B) = k + t$, implying the surjectivity of ev. So, we have $\mathcal{B}$-privacy with uniformity. □

Note that by Theorem 3.6, the equation system of Theorem 3.8 has a solution for $\Gamma = (\mathcal{A}, \mathcal{B})$, if there is an $s \in \mathbb{Z}$ such that

$$h_F > \sum_{i=1}^{|\max \mathcal{B}|} E_{s_i} + \sum_{i=1}^{|\min \mathcal{A}|} E_{r_i} \cdot |J_F[d]|, \tag{3.2}$$

where E_i is the number of effective divisors of degree i, $s_i = 2g - 2 - s + |B_i| + k$, $r_i = ds - |A_i|$ and $\max \mathcal{B} = \{B_1, \ldots, B_{|\max \mathcal{B}|}\}$, $\min \mathcal{A} = \{A_1, \ldots, A_{|\min \mathcal{A}|}\}$. For the threshold case of $\Gamma = \Gamma(t, n - t)$, this simplifies to

$$h_F > \binom{n}{t} \Big(E_{s'_1} + E_{r'_1} \cdot |J_F[d]| \Big), \tag{3.3}$$

where $s'_i = 2g - 2 - s + t + k$, $r'_i = ds - n + t$, cf. [CCX12b, Corollary 4.12].

We have the following bound on the number of effective divisors of a given degree.

Proposition 3.2 ([CCX12b, Proposition 3.4]). *Let F be an algebraic function field with exact constant field $\mathbb{F}_q$. For $d \in \mathbb{N}$, let E_d denote the number of effective divisors of degree d in* $\mathrm{Div}(F)$. *Suppose $g := g_F \geq 1$, then, for any $d \in \mathbb{N}$, with $d \leq g-1$, we have*

$$E_d \leq \frac{g \cdot h_F}{q^{g-d-1}(\sqrt{q}-1)^2}.$$

Using this, we can simplify (3.2) in the case of $g_F \geq 1$ and r_i, $s_i \leq g_F - 1$ to

$$\begin{aligned} 1 &> \sum_{i=1}^{|\max \mathcal{B}|} \frac{g}{q^{g-s_i-1}(\sqrt{q}-1)^2} + \sum_{i=1}^{|\min \mathcal{A}|} \frac{g}{q^{g-r_i-1}(\sqrt{q}-1)^2} \cdot |J_F[d]| \\ &= \frac{g}{q^{g-1}(\sqrt{q}-1)^2}\left(q^{2g-2-s+k}\sum_{i=1}^{|\max \mathcal{B}|} q^{|B_i|} + q^{ds}|J_F[d]| \sum_{i=1}^{|\min \mathcal{A}|} q^{-|A_i|}\right). \end{aligned}$$

Establishing this inequality in a family of global function fields with beneficial properties, allows one then to prove the following theorem.

Theorem 3.9 ([CCX12b, Theorem 4.13]). *Let $\mathbb{F}_q$ be a finite field and $d \in \mathbb{Z}_{\geq 2}$. If there exists $0 < a \leq A(q)$, such that $a > 1 + J_d(q,a)$, then there is an infinite family of $(n,t,d,n-t)$-arithmetic secret sharing schemes for $\mathbb{F}_q^k$ over $\mathbb{F}_q$ with t-uniformity where n is unbounded, $k = \Omega(n)$ and $t = \Omega(n)$.*

As the existence of such an a has been shown in Corollary 3.2 all in all it follows that there is an infinite family of $(n,t,d,n-t)$-arithmetic secret sharing schemes for $\mathbb{F}_q^k$ over $\mathbb{F}_q$ with t-uniformity where n is unbounded, $k = \Omega(n)$ and $t = \Omega(n)$.

4 Conclusion

As far as we know, this thesis provides the first *computational* secret sharing scheme with homomorphic properties, such as *strong multiplicativity.* It works by a combination of homomorphic encryption and sharing and is introduced in two variants. In the first variant, the secret is first encrypted and shared afterwards, while in the second, it is first shared and the shares are then encrypted. For the first variant to work, we have to execute the share map as a circuit by using the homomorphic evaluation of the encryption scheme, suggesting that it is less preferable, as it starts with some encryption noise already. In contrast, the encryption of the shares in the second version is fresh; it has therefore less noise and works with larger circuits, if a leveled fully homomorphic encryption scheme is used. Noting this, the second variant has the only caveat that the special share map on the share spaces as in the definition of strongly multiplicative secret sharing schemes, is not as directly obtained as the one in the first version.

Note that typically the encryption map of a fully homomorphic encryption scheme is not linear—in contrast to its decryption, provided that we can execute arbitrary circuits—and therefore our scheme satisfies only a weaker version of linear secret sharing as usually found in the literature. We add to this variant a discussion on secrecy guarantees after a calculation on the shares, in a game-based manner typical for the field of provable security. With this, it is easy to see the suitability for passively secure multiparty computation, however, for most constructions of verifiable secret sharing and actively secure MPC it is usually presumed that the share map is linear, e. g., as in [CDM00; FM02; C^+03]. We point the reader to a protocol of Maurer [M03], which does not make use of these properties, and should work well together with our homomorphic CSS scheme.

Concerning Chapter 3, we gave an overview of a part of the literature of secret sharing schemes using algebraic geometric codes, with focus on a paper of Cascudo, Cramer, and Xing [CCX12b]. Here, we highlighted their construction of infinite class field towers with the theorem of Golod–Šafarevič and their use of Riemann–Roch systems of equations to construct asymptotically good families of arithmetic secret sharing schemes.

Future Work

Concerning future work we would like to summarize the open questions raised during the course of writing the thesis. This includes a yet missing formal proof of suitability for actively secure MPC of our homomorphic CSS scheme. Moreover, a formal analysis of the information rate of our scheme and the communication complexity for its use in VSS and MPC protocols, would provide a nice addition to the thesis. It would be interesting whether there are more powerful criteria for secrecy against $\mathcal{C}$-homorophic $\mathcal{B}$-privacy adversaries. As we have noted in the text, we could have devised the scheme in a way to work on power sets of the key space instead of the algebra generated by its elements, yielding less information which is not needed for the correct reconstruction. For this, it would be necessary to analyze, whether all statements, including the suitability for VSS and MPC protocols, still hold if the reconstruction is a linear map of semimodules over semirings, see Remark 2.1. Moreover, we would have liked to include a short discussion on circuit privacy of multikey fully homomorphic encryption schemes.

For the second part it would be interesting to generalize Theorem 3.5 to other values of torsion besides 2. For this, the statement might follow from more general arguments as in [NSW13, (10.10.3)], which would need to be adapted to our setting of global function fields where we have to additionally keep an eye on the growth of the genus. During the last section, we gave indications of the generalization of the notion of a codex to non-threshold access structures and it would be interesting to adapt Theorem 3.9 to this setting.

Glossary of Symbols

(f)	principal divisor of $f \in F^{\times}$. 38
(ω)	canonical divisor. 39
$[K:F]$	vector space dimension, degree of a field extension. 38
$[a]_i$	commitment of player P_i containing value a. 13
$\lVert\cdot\rVert$	maximum norm of polynomial coefficients of R. 20
$\lvert\cdot\rvert$	length of a string, cardinality of a set, absolute value. 1
$\mathcal{A}$	set of qualified players who can reconstruct the secret message. 2
$A(\mathcal{F})$	Ihara limit of a family $\mathcal{F}$ of function fields. 41
$A(q)$	Ihara limit of all families of function fields over $\mathbb{F}_q$. 41
$\mathcal{A}_1 \sqcup \mathcal{A}_2$	union of set systems $\mathcal{A}_1$ and $\mathcal{A}_2$. 7
$\overline{\mathcal{A}}$	complement of a set system $\mathcal{A}$. 2
$\mathbf{Adv}^{\text{IND-CPA}}_{\mathsf{HE},A}(\kappa)$	advantage of an IND-CPA adversary A. 19
$\mathbf{Adv}^{\text{priv}}_{\Sigma,A}(\kappa)$	advantage of privacy adversary A in Σ. 2, 3
$\mathbf{Adv}^{\mathcal{C}\text{-priv}}_{\Sigma,A}(\kappa)$	advantage of a $\mathcal{C}$-homomorphic privacy adversary A. 9
$\mathbf{Adv}^{\text{rec}}_{\Sigma,A}(\kappa)$	advantage of reconstruction adversary A in Σ. 3
$\mathbf{Adv}^{\mathcal{C}\text{-rec}}_{\Sigma,A}(\kappa)$	advantage of a $\mathcal{C}$-homomorphic reconstruction adversary A. 9
A_F	adèle ring of F. 39
$A_F(D)$	set of all $(a_{\mathfrak{p}}) \in A_F$, satisfying $v_{\mathfrak{p}}(a_{\mathfrak{p}}) \geq -n(\mathfrak{p})$ for all primes $\mathfrak{p}$. 39
$\mathcal{B}$	set of unqualified players ignorant about the secret. 2
$\mathcal{C}$	class of algebraic circuits. 8
$C_L(D, G)$	Goppa code with divisors D, G. 40
$\mathrm{Cl}(F)$	class group of F, defined as $\mathrm{Div}(F)/\operatorname{Prin}(F)$. 38
$\mathcal{C}^{\text{lin}}$	class of all linear circuits. 8
$\mathrm{Cl}_T(F)$	T-class group of F. 44
$\mathsf{corrupt}(\boldsymbol{s}, i)$	corruption oracle, returns share of player P_i. 3
$\mathcal{C}_{t\to l}$	class of all t-ary circuits with l output nodes. 8
$\mathcal{C}^{\leq L}$	class of all circuits with polynomial size and depth $\leq L$. 8
$d(C)$	mimimal distance of a code C. 40
$d(G)$	rank of a pro-p-group. 42
$d(\boldsymbol{x}, \boldsymbol{y})$	Hamming distance of two vectors $\boldsymbol{x}$, $\boldsymbol{y}$. 40
$\deg(\mathfrak{p})$	degree of a prime $\mathfrak{p}$. 38
$\mathrm{Div}^{(d)}(F)$	set of divisors of degree d on F. 38
$\mathrm{Div}(F)$	group of divisors on F. 38
$D_{q,w}(\cdot)$	decomposition map. 21
ek	evaluation key to compute homomorphically on ciphertexts. 17
E_r	number of effective divisors of degree r. 50

$\mathsf{EvalKey}(\boldsymbol{s})$	map returning the evaluation key attached to a share vector. 9
F/k	algebraic function field over a constant field k. 37
$F_S^T(p)$	maximal p-extension of F unramified outside S and completely split at every prime of T. 44
$F_\infty^T(p)$	union of all extensions in a (T, p)-Hilbert class field tower. 44
g_F	genus of an algebraic function field. 39
HE	homomorphic encryption scheme. 17, 27, 36
h_F	class number of F. 50
id_X	identity map on X. 5
IDA	information dispersal algorithm. 27, 36
$\mathrm{im}(\cdot)$	image space of a map. 4
J_F	zero divisor class group of F. 38
$J_F[r]$	r-torsion elements in the zero divisor class group of F. 50
$J_r(q, A)$	least possible r-torsion of a family of function fields for which the Ihara limit exceeds a given value A. 51
$J_r(\mathcal{F})$	r-torsion limit of a family $\mathcal{F}$. 51
k^{ab}	abelian closure of a field k. 44
k^{alg}	algebraic closure of a field k. 43
KSS	key secret sharing scheme. 27, 36
$L(D)$	Riemann–Roch space of divisor D. 39
$\ell(D)$	$\mathbb{F}_q$-dimension of the Riemann–Roch space $L(D)$. 39
$\mathsf{Lift}(C)$	map lifting a circuit C from the message to the share space. 9
$\mathcal{M}$	message space of a secret sharing scheme. 1, 4
$\max \mathcal{B}$	max-terms of $\mathcal{B}$. 2
$\min \mathcal{A}$	min-terms of $\mathcal{A}$. 2
$\mathbb{N}$	natural numbers, including zero. 1
$N(F)$	number of primes of degree one of F. 41
$\mathrm{negl}(\kappa)$	negligible function, less than $1/p(\kappa)$, for any polynomial p. 1
$N_q(g)$	maximal number of primes of degree one for any function field $F/\mathbb{F}_q$ with genus g. 41
$\mathcal{O}$	discrete valuation ring of a prime. 37
$\mathcal{O}_{F,T}$	ring of T integers. 44
P	set of n players, $P \coloneqq \{P_1, \dots, P_n\}$. 1
$\mathfrak{p}$	a prime of an algebraic function field. 38
$\mathcal{P}(X)$	power set of a set X. 2
$P_{q,w}(\cdot)$	power-of-w map. 21
$\mathrm{Prin}(F)$	group of principal divisors on F. 38
Q^3	Q^3 access structure. 7
R	ring $\mathbb{Z}[X]/\Phi_d(X)$ used in NTRU-based encryption schemes. 20
$\mathcal{R}$	randomness space. 5
$r(G)$	relation rank of a pro-p-group. 42
$R[T]$	polynomial ring with coefficients in R and variable T. 6
Rec	reconstruction map $\mathsf{Rec}\colon \prod_{i=1}^n (\mathcal{S}_i \cup \{\Diamond\}) \times \{0, \dots, n\} \to \mathcal{M} \cup \{\bot\}$ of an SSS. 1

$r_t(x)$	representation of x modulo t. 21
$R^\times$	invertible elements (units) of R. 6
$R\langle X\rangle$	R-algebra generated on a set X. 27
$\mathcal{S}$	share space of a secret sharing scheme, $\mathcal{S}_1 \times \cdots \times \mathcal{S}_n$. 4
$\boldsymbol{s} * \boldsymbol{s}'$	multiplication of shares, including resharing. 6
Sh	share map $\mathsf{Sh}\colon \mathcal{M} \to \mathcal{S}$ of a secret sharing scheme. 1
Sh'_i	share function on the share space $\mathcal{S}_i$ of P_i. 7, 33
$\boldsymbol{s}_i \circledast_i \boldsymbol{s}'_i$	multiplication in share space $\mathcal{S}_i$ of P_i. 6, 33
$\boldsymbol{s}_T$	vector $\boldsymbol{s}$ restricted to index set T. 1
$\boldsymbol{s}_{\overline{T}} \sqcup \boldsymbol{s}'_T$	the vector $\boldsymbol{x}$ with $\boldsymbol{x}_i = \boldsymbol{s}_i$ if $i \notin T$ and $\boldsymbol{x}_i = \boldsymbol{s}'_i$ if $i \in T$. 5
$v_{\mathfrak{p}}(\cdot)$	valuation of a prime $\mathfrak{p}$. 38
$x \leftarrow \chi$	sampling x according to probability distribution χ. 1, 20
Γ	access structure specifying qualified/unqualified player coalitions. 2
$\Gamma(t, r)$	threshold access structure. 2
Δ	equals $\lfloor q/t \rfloor$. 21
Σ	A secret sharing scheme, $\Sigma = (\mathsf{Sh}, \mathsf{Rec})$. 1, 3, 4
$\Phi_d(X)$	d-th cyclotomic polynomial. 20
Ω	space of Weil differentials on an algebraic function field. 39
δ	$\sup_{a,b\in R}\left(\frac{\|ab\|}{\|a\|\|b\|}\right)$. 20
κ	the security parameter $\kappa \in \mathbb{N}$. 2
μ_p	pth roots of unity. 46
$\pi_0(c)$	$\pi_0\colon \mathcal{M} \times \mathbb{F}_q^n \to \mathcal{M}$, projection of $\boldsymbol{x}$ to its $\mathcal{M}$-component. 51
$\pi_A(c)$	$\pi_A\colon \mathcal{M} \times \mathbb{F}_q^n \to \mathbb{F}_q^{\lvert A\rvert}$, projection of $\boldsymbol{x}$ to the coordinates of A.. 51
φ	ring homomorphism of LSSS, also Euler phi function. 4, 7
$\varphi^*(\mathcal{M})$	restriction of scalars of module $\mathcal{M}$. 5
ψ	key reduction in CSS, $\psi\colon R\langle\mathcal{K}\rangle \to \mathcal{P}(\mathcal{K})$. 28
ω	Weil differential ω. 39

Bibliography

[A$^+$12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. "Multiparty Computation with Low Communication, Computation and Interaction via Threshold FHE". In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by David Pointcheval and Thomas Johansson. Vol. 7237. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 483–501. DOI: 10.1007/978-3-642-29011-4_29.

[B$^+$13] Joppe W. Bos, Kristin Lauter, Jake Loftus, and Michael Naehrig. *Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme*. 2013. Cryptology ePrint Archive, Report 2013/075.

[B11] Amos Beimel. "Secret-Sharing Schemes: A Survey". In: *Coding and Cryptology*. Ed. by Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing. Vol. 6639. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 11–46. DOI: 10.1007/978-3-642-20901-7_2.

[B12] Zvika Brakerski. "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP". In: *Advances in Cryptology – CRYPTO 2012*. Ed. by Reihaneh Safavi-Naini and Ran Canetti. Vol. 7417. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 868–886. DOI: 10.1007/978-3-642-32009-5_50.

[B79] George R. Blakley. "Safeguarding cryptographic keys". In: *Managing Requirements Knowledge, International Workshop on* (1979), p. 313. DOI: 10.1109/AFIPS.1979.98.

[B96] Amos Beimel. "Secure Schemes for Secret Sharing and Key Distribution". Ph. D. thesis. Technion – Israel Institute of Technology, 1996. URL: http://www.cs.bgu.ac.il/~beimel/Papers/thesis.ps.

[BC95] Philippe Béguin and Antonella Cresti. "General Short Computational Secret Sharing Schemes". In: *Advances in Cryptology – EUROCRYPT '95*. Ed. by Louis C. Guillou and Jean-Jacques Quisquater. Vol. 921. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1995, pp. 194–208. DOI: 10.1007/3-540-49264-X_16.

[BGS05] Juscelino Bezerra, Arnaldo Garcia, and Henning Stichtenoth. "An explicit tower of function fields over cubic finite fields and Zink's lower bound". In: *Journal für die Reine und Angewandte Mathematik* 589 (2005), pp. 159–199. DOI: 10.1515/crll.2005.2005.589.159.

[BGV12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. "(Leveled) Fully Homomorphic Encryption Without Bootstrapping". In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ITCS '12. Cambridge, Massachusetts: ACM, 2012, pp. 309–325. DOI: 10.1145/2090236.2090262.

[BKP11] Michael Backes, Aniket Kate, and Arpita Patra. "Computational Verifiable Secret Sharing Revisited". In: *Advances in Cryptology – ASIACRYPT 2011*. Ed. by Dong Hoon Lee and Xiaoyun Wang. Vol. 7073. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 590–609. DOI: 10.1007/978-3-642-25385-0_32.

[BL90] Josh C. Benaloh and Jerry Leichter. "Generalized Secret Sharing and Monotone Functions". In: *Proceedings of the 8th Annual International Cryptology Conference on Advances in Cryptology*. CRYPTO '88. London, UK: Springer-Verlag, 1990, pp. 27–35. URL: http://dl.acm.org/citation.cfm?id=646753.704890.

[BR07] Mihir Bellare and Phillip Rogaway. "Robust Computational Secret Sharing and a Unified Account of Classical Secret-sharing Goals". In: *Proceedings of the 14th ACM Conference on Computer and Communications Security*. CCS '07. Alexandria, Virginia, USA: ACM, 2007, pp. 172–184. DOI: 10.1145/1315245.1315268.

[BV11a] Zvika Brakerski and Vinod Vaikuntanathan. "Efficient Fully Homomorphic Encryption from (Standard) LWE". In: *Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*. FOCS '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 97–106. DOI: 10.1109/FOCS.2011.12.

[BV11b] Zvika Brakerski and Vinod Vaikuntanathan. "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages". In: *Advances in Cryptology – CRYPTO 2011*. Ed. by Phillip Rogaway. Vol. 6841. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 505–524. DOI: 10.1007/978-3-642-22792-9_29.

[C$^+$03] Ronald Cramer, Serge Fehr, Yuval Ishai, and Eyal Kushilevitz. "Efficient Multi-party Computation over Rings". In: *Advances in Cryptology – EUROCRYPT 2003*. Ed. by Eli Biham. Vol. 2656. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 596–613. DOI: 10.1007/3-540-39200-9_37.

[C$^+$08] Hao Chen, Ronald Cramer, Robbert Haan, and Ignacio Cascudo. "Strongly Multiplicative Ramp Schemes from High Degree Rational Points on Curves". In: *Advances in Cryptology – EUROCRYPT 2008*. Ed. by Nigel Smart. Vol. 4965. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 451–470. DOI: 10.1007/978-3-540-78967-3_26.

[C$^+$09] Ignacio Cascudo, Hao Chen, Ronald Cramer, and Chaoping Xing. "Asymptotically Good Ideal Linear Secret Sharing with Strong Multiplication over Any Fixed Finite Field". In: *Advances in Cryptology – CRYPTO 2009*. Ed. by Shai Halevi. Vol. 5677. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pp. 466–486. DOI: 10.1007/978-3-642-03356-8_28.

[C01] Ran Canetti. "Universally Composable Security: A New Paradigm for Cryptographic Protocols". In: *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*. FOCS '01. Washington, DC, USA: IEEE Computer Society, 2001, 136sqq. URL: http://dl.acm.org/citation.cfm?id=874063.875553.

[C97] László Csirmaz. "The Size of a Share Must Be Large". In: *Journal of Cryptology* 10.4 (1997), pp. 223–231. DOI: 10.1007/s001459900029.

[CC06] Hao Chen and Ronald Cramer. "Algebraic Geometric Secret Sharing Schemes and Secure Multi-Party Computations over Small Fields". In: *Advances in Cryptology – CRYPTO 2006*. Ed. by Cynthia Dwork. Vol. 4117. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 521–536. DOI: 10.1007/11818175_31.

[CCX11] Ignacio Cascudo, Ronald Cramer, and Chaoping Xing. "The Torsion-Limit for Algebraic Function Fields and Its Application to Arithmetic Secret Sharing". In: *Advances in Cryptology – CRYPTO 2011*. Ed. by Phillip Rogaway. Vol. 6841. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 685–705. DOI: 10.1007/978-3-642-22792-9_39.

[CCX12a] Ignacio Cascudo, Ronald Cramer, and Chaoping Xing. *The Arithmetic Codex*. 2012. Cryptology ePrint Archive, Report 2012/388.

[CCX12b] Ignacio Cascudo, Ronald Cramer, and Chaoping Xing. "Torsion Limits and Riemann–Roch Systems for Function Fields and Applications". In: *ArXiv e-prints* (2012). ID: 1207.2936 [math.AG].

[CD05] Ronald Cramer and Ivan Damgård. "Multiparty Computation, an Introduction". In: *Contemporary Cryptology*. Advanced Courses in Mathematics – CRM Barcelona. Birkhäuser Basel, 2005, pp. 41–87. DOI: 10.1007/3-7643-7394-6_2.

[CDI05] Ronald Cramer, Ivan Damgård, and Yuval Ishai. "Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation". In: *Theory of Cryptography*. Ed. by Joe Kilian. Vol. 3378. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 342–362. DOI: 10.1007/978-3-540-30576-7_19.

[CDM00] Ronald Cramer, Ivan Damgård, and Ueli Maurer. "General Secure Multiparty Computation from any Linear Secret-Sharing Scheme". In: *Advances in Cryptology – EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2000, pp. 316–334. DOI: 10.1007/3-540-45539-6_22.

[CF02] Ronald Cramer and Serge Fehr. “Optimal Black-Box Secret Sharing over Arbitrary Abelian Groups”. In: *Advances in Cryptology – CRYPTO 2002*. Ed. by Moti Yung. Vol. 2442. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 272–287. DOI: 10.1007/3-540-45708-9_18.

[D08] Iwan M. Duursma. “Algebraic Geometry Codes: General Theory”. In: *Advances in Algebraic Geometry Codes*. Ed. by Edgar Martínez-Moro, Carlos Munuera, and Diego Ruano. Vol. 5. Series on Coding Theory and Cryptology. Singapore: World Scientific Publishing Co. Pte. Ltd., 2008. Chap. 1, pp. 1–48. DOI: 10.1142/9789812794017_0001. URL: https://www.worldscientific.com/doi/suppl/10.1142/6767/suppl_file/6767_chap01.pdf.

[FM02] Serge Fehr and Ueli Maurer. “Linear VSS and Distributed Commitments Based on Secret Sharing and Pairwise Checks”. In: *Advances in Cryptology – CRYPTO 2002*. Ed. by Yung Moti. Vol. 2442. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 565–580. DOI: 10.1007/3-540-45708-9_36.

[FV12] Junfeng Fan and Frederik Vercauteren. *Somewhat Practical Fully Homomorphic Encryption*. 2012. Cryptology ePrint Archive, Report 2012/144.

[G01] Ana Gàl. “A characterization of span program size and improved lower bounds for monotone span programs”. In: *Computational Complexity* 10.4 (2001), pp. 277–296. DOI: 10.1007/s000370100001.

[G09] Craig Gentry. “A Fully Homomorphic Encryption Scheme”. Ph. D. thesis. Stanford, CA, USA: Stanford University, 2009.

[H$^+$11] Martin Hirt, Christoph Lucas, Ueli Maurer, and Dominik Raub. “Graceful Degradation in Multi-Party Computation (Extended Abstract)”. In: *Information Theoretic Security*. Ed. by Serge Fehr. Vol. 6673. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 163–180. DOI: 10.1007/978-3-642-20728-0_15.

[H02] Florian Hess. “Computing Riemann–Roch Spaces in Algebraic Function Fields and Related Topics”. In: *Journal of Symbolic Computation* 33.4 (2002), pp. 425–445. DOI: 10.1006/jsco.2001.0513.

[HM13] Florian Hess and Maike Massierer. “Class Field Theory for Global Function Fields”. In: *ArXiv e-prints* (2013). ID: 1304.2131 [math.NT].

[I82] Yasutaka Ihara. “Some remarks on the number of rational points of algebratic curves over finite fields”. In: *Journal of the Faculty of Science, the University of Tokyo. Sect. 1A, Mathematics* 28.3 (1982), pp. 721–724. URL: http://ci.nii.ac.jp/naid/120000869903/en/.

[ISN89] Mitsuru Ito, Akira Saito, and Takao Nishizeki. “Secret sharing scheme realizing general access structure”. In: *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)* 72.9 (1989), pp. 56–64. DOI: 10.1002/ecjc.4430720906.

[J13] A. E. de Jonge. “Bounds on the parameters of arithmetic codices”. Bachelor thesis. Mathematical Institute, Leiden University, 2013.

[K$^+$13] Ryo Kikuchi, Koji Chida, Dai Ikarashi, Koki Hamada, and Katsumi Takahashi. “Secret Sharing Schemes with Conversion Protocol to Achieve Short Share-Size and Extendibility to Multiparty Computation”. In: *Information Security and Privacy*. Ed. by Colin Boyd and Leonie Simpson. Vol. 7959. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 419–434. DOI: 10.1007/978-3-642-39059-3_29.

[K70] Helmut Koch. *Galoissche Theorie der p-Erweiterungen*. Vol. 10. Mathematische Monographien. VEB Deutscher Verlag der Wissenschaften, Berlin, 1970.

[K94] Hugo Krawczyk. “Secret Sharing Made Short”. In: *Advances in Cryptology – CRYPTO ’93*. Ed. by Douglas R. Stinson. Vol. 773. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1994, pp. 136–146. DOI: 10.1007/3-540-48329-2_12.

[KW93] Mauricio Karchmer and Avi Wigderson. “On span programs”. In: *Structure in Complexity Theory Conference, Proceedings of the 8th Annual*. 1993, pp. 102–111. DOI: 10.1109/SCT.1993.336536.

[LM06] Vadim Lyubashevsky and Daniele Micciancio. “Generalized Compact Knapsacks Are Collision Resistant”. In: *Automata, Languages and Programming*. Ed. by Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener. Vol. 4052. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 144–155. DOI: 10.1007/11787006_13.

[LO13] Joshua Lampkins and Rafail Ostrovsky. *Communication-Efficient MPC for General Adversary Structures*. 2013. Cryptology ePrint Archive, Report 2013/640.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. “On Ideal Lattices and Learning with Errors over Rings”. In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[LTV12] Adriana López-Alt, Eran Tromer, and Vinod Vaikuntanathan. “On-the-fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption”. In: *Proceedings of the 44th Symposium on Theory of Computing*. STOC ’12. New York, NY, USA: ACM, 2012, pp. 1219–1234. DOI: 10.1145/2213977.2214086.

[M03] Ueli Maurer. “Secure Multi-party Computation Made Simple”. In: *Security in Communication Networks*. Ed. by Stelvio Cimato, Giuseppe Persiano, and Clemente Galdi. Vol. 2576. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 14–28. DOI: 10.1007/3-540-36413-7_2.

[N03] Jesper Buus Nielsen. “On protocol security in the cryptographic model”. Ph. D. thesis. BRICS, Computer Science Department, University of Aarhus, 2003.

[N92] Jürgen Neukirch. *Algebraische Zahlentheorie*. Springer Berlin Heidelberg, 1992.

[NSW13] Jürgen Neukirch, Alexander Schmidt, and Kay Wingberg. *Cohomology of Number Fields*. Sec. Ed., corr. sec. print. Vol. 323. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2013. URL: https://www.mathi.uni-heidelberg.de/~schmidt/NSW2e/.

[NX01] Harald Niederreiter and Chaoping Xing. *Rational Points on Curves over Finite Fields: Theory and Applications*. Vol. 288. London Mathematical Society Lecture Note Series. Cambridge, UK: Cambridge University Press, 2001.

[P12] Carles Padró. *Lecture Notes in Secret Sharing*. 2012. Cryptology ePrint Archive, Report 2012/674.

[PK09] Abhishek Parakh and Subhash Kak. “Space Efficient Secret Sharing: A Recursive Approach”. In: *ArXiv e-prints* (2009). ID: 0901.4814 [cs.CR].

[R02] Michael Rosen. *Number Theory in Function Fields*. Graduate Texts in Mathematics 210. Springer New York, 2002.

[R89] Michael O. Rabin. “Efficient Dispersal of Information for Security, Load Balancing, and Fault Tolerance”. In: *Journal of the ACM* 36.2 (1989), pp. 335–348. DOI: 10.1145/62044.62050.

[RP11] Jason K. Resch and James S. Plank. “AONT-RS: Blending Security and Performance in Dispersed Storage Systems”. In: *Proceedings of the 9th USENIX Conference on File and Storage Technologies*. FAST ’11. San Jose, California: USENIX Association, 2011, pp. 1–12. URL: http://dl.acm.org/citation.cfm?id=1960475.1960489.

[S79] Adi Shamir. “How to Share a Secret”. In: *Communications of the ACM* 22.11 (1979), pp. 612–613. DOI: 10.1145/359168.359176.

[S93] Henning Stichtenoth. *Algebraic Function Fields and Codes*. Springer Berlin Heidelberg, 1993.

[SS11] Damien Stehlé and Ron Steinfeld. “Making NTRU as Secure as Worst-Case Problems over Ideal Lattices”. In: *Advances in Cryptology – EUROCRYPT 2011*. Ed. by Kenneth G. Paterson. Vol. 6632. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 27–47. DOI: 10.1007/978-3-642-20465-4_4.

[T10] Oliver Thomas. *p-Klassengruppen und p-Klassenkörpertürme. Zahlentheoretisches Seminar: Klassenkörpertürme – der Satz von Golod–Šafarevič*. Vortrag 12. Heidelberg, 2010. URL: https://www.mathi.uni-heidelberg.de/~bartels/Alt/Klassenkoerpertuerme10/Vortrag12.pdf. unpublished.

[TVW13] Stephen R. Tate, Roopa Vishwanathan, and Scott Weeks. *Encrypted Secret Sharing and Analysis by Plaintext Randomization.* 2013. Cryptology ePrint Archive, Report 2013/264.

[V$^+$03] Vaikuntanathan Vinod, Arvind Narayanan, K. Srinathan, C. Pandu Rangan, and Kwangjo Kim. “On the Power of Computational Secret Sharing”. In: *Progress in Cryptology – INDOCRYPT 2003*. Ed. by Thomas Johansson and Subhamoy Maitra. Vol. 2904. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2003, pp. 162–176. DOI: 10.1007/978-3-540-24582-7_12.

[VD83] Serge G. Vlăduţ and Vladimir G. Drinfeld. “Number of points of an algebraic curve”. In: *Functional Analysis and Its Applications* 17.1 (1983), pp. 53–54. DOI: 10.1007/BF01083182.

[Z85] Thomas Zink. “Degeneration of Shimura surfaces and a problem in coding theory”. In: *Fundamentals of Computation Theory*. Ed. by Lothar Budach. Vol. 199. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1985, pp. 503–511. DOI: 10.1007/BFb0028834.