Span Programs over Rings and How to Share a Secret from a Module

Master Thesis WS 97/98

Serge Fehr

Prof. Dr. U. Maurer
Institute for Theoretical Computer Science
ETH Zurich

Supervised by Dr. R. Cramer

Contents

Chapter 1

Introduction

Most cryptographic schemes rely on a key that has to be kept secret. For example in a digital signature scheme, we assume that the secret key to sign a message is known only to one person or to a group of people where each participant can sign messages in the name of the whole group.

However, there are several disadvantages to this assumption. When, for instance, the digital signature scheme is being used by a company to sign checks or contracts in the name of the company, then it must completely trust the person who knows the secret key not to lose it and not to misuse it either. When there is a group of people where each knows the secret key, then the risk of a security breach, both by accident or on purpose, increases with the size of the group.

It would be nice if the company could apply a scheme to share the secret key among a group of people in such a way that only specified subsets of the people in that group, called *authorized* subsets, can establish the key, whereas the *unauthorized* subsets cannot. Even though this would not solve the above signature problem satisfactory, it is a main primitive for what it is called *signature sharing schemes*. Schemes to realize this primitive of sharing a secret are called *secret sharing schemes*. One could think, for instance, of a scheme that requires that out of l people, any t can construct the key while fewer people have no information about it. Such a scheme, called (t, l)-*threshold scheme*, was presented by Shamir in [21].

In this scheme, the *dealer* generates a random polynomial $f(X)$ of degree at most $t-1$ over a finite field (whose size is greater than l) with the only restriction that $f(0) = s$, the secret, and gives each *player* i the share $f(i)$. Using Lagrange interpolation, any t players can compute s from their shares. But a group of less then t shareholders cannot; in fact, their shares give away no information about the secret at all.

There are three major restrictions in this scheme. The first one is, of course, that it only works if the authorized subsets are defined by such a threshold t, but it cannot be used for other, more general *access structures*. An access structure describes which subgroups are authorized and hence should be able to reconstruct the secret and which are not. Indeed, one could think of a company where to sign checks, there should be present, for example, at least two directors, a director and three vice-directors or five vice-directors. In this case, the Shamir-scheme cannot be used, at least not in a straight forward way.
The second restriction is that it is not secure against active adversaries. Some faulty players can sabotage the reconstruction by sending incorrect shares instead of the ones received from the dealer. Or a faulty dealer can distribute random elements instead of shares of a correctly shared secret. A scheme which is secure against faulty players is called *robust*, if it is secure against a faulty dealer and faulty players, it is called *verifiable*.
The third restriction in the Shamir-scheme is that the secret s has to be an element of a finite field. But, for instance, in the RSA signature scheme [20], the secret key is an element of $\mathbb{Z}_{\varphi(n)}$, an Abelian group. One way to share a secret from this group would be to see $\mathbb{Z}_{\varphi(n)}$ as a subset of $\mathbb{Z}_p$, where p is a prime larger than $\varphi(n)$, and to share the secret $s \in \mathbb{Z}_{\varphi(n)}$ as an element from the finite field $\mathbb{Z}_p$. The problem with this method is that it is not *homomorphic*. Namely, if s_i and s'_i, $i \in \mathcal{P}$, are the shares of the secret s and s', respectively, then $s_i + s'_i$, $i \in \mathcal{P}$, are not necessarily shares of $s + s'$. But as this is needed in many applications of secret sharing schemes, e. g. multiparty computations [1], this is an important property.

There exist many papers that deal with the first restriction, i. e. with secret sharing schemes for (more) general access structures than threshold access structures. We will give a quick and incomplete overview.
The first to suggest a method for sharing a secret for an arbitrary access structure were Ito, Saito and Nishizeki [13]. Their main idea was to apply the

(m,m)-threshold scheme independently for every minimal authorized subset A where $m = |A|$, the cardinality of A. The problem herewith is of course that the number of minimal authorized subsets is, in general, exponential in the size of the group of people.
Completely different from this method is Brickell's vector space construction [3]. For this we have to assume that for the access structure there exists a so called *vector distribution function*. This is a function that identifies every player $i \in \mathcal{P}$ with a vector $\mathbf{v}_i \in F^e$, where F is a field, such that a set of players $A \subseteq \mathcal{P}$ is authorized if and only if the first unit vector $\mathbf{e}_1 = (1, 0, \ldots, 0) \in F^e$ is in the subspace span$\{\mathbf{v}_i \mid i \in A\}$. If every player i gets the share $s_i = \langle \mathbf{v}_i, \mathbf{x} \rangle$, where $\mathbf{x} = (s, x_2, \ldots, x_e)$ with $s \in F$ the secret and the other coordinated chosen at random, then an authorized subset A can compute the secret as $s = \sum_{i \in A} \lambda_i s_i$, where the coefficients λ_i fulfil $\sum_{i \in A} \lambda_i \mathbf{v}_i = \mathbf{e}_1$. Further, it can be shown that an unauthorized subset gets no information about the secret.
In [2] Benaloh and Leichter presented secret sharing schemes for general access structures based on monotone formulae.
Finally, Karchmer and Wigderson [14] introduced *span programs* and showed how monotone span programs give rise to secret sharing schemes for general access structures. A monotone span program identifies every player $i \in \mathcal{P}$ with some vectors in F^e which we see as columns of the matrix M_i. If a set A of players is authorized if and only if $\mathbf{e}_1 = (1, 0, \ldots, 0) \in F^e$ lies in the span of the vectors of the players in A, then the monotone span program is said to *compute* the access structure. Note that the special case of a monotone span program where every player is identified with only one vector coincides with Brickell's vector distribution function. A secret $s \in F$ can now be shared by distributing the shares $\mathbf{s}_i = M_i \mathbf{x}$, where $\mathbf{x} \in F^e$ is chosen at random with the only restriction that the first coordinate is the secret s. It can be reconstructed by an authorized subset in a similar way as in Brickell's vector space construction. It can be shown that not only Brickell's vector space construction but all the schemes mentioned so far are special cases of this scheme based on span programs.

In [6] Chor *et al.* introduced robust and verifiable secret sharing schemes, schemes secure against faulty players and a faulty dealer plus faulty players. Their methods are based on some intractability assumption (such as "factoring large integers is infeasible"), whereas the verifiable schemes by Ben-Or, Goldwasser and Wigderson [1] and by Chaum, Crépeau and Damgård are unconditionally secure. Cramer, Damgård and Maurer [8] presented an (unconditionally secure) verifiable scheme for general access structures, based on Karchmer and Wigderson's span program construction [14]. They also introduced span programs with multiplication as a primitive for general secure multiparty computations.

Desmedt and Frankel showed in [10] how to modify the Shamir-scheme to a homomorphic threshold scheme which shares a secret from a finite Abelian group G instead of a finite field. Their idea was to see G as a module over $\mathbb{Z}_e$, where e is the exponent of G, or over $\mathbb{Z}$. A secret $s \in G$ can now be shared analogue to Shamir's scheme. The dealer chooses a random polynomial $f(X)$ of degree at most $t-1$ with coefficients in G with the only restriction that $f(0) = s$, and gives each player i the share $f(\omega_i)$, where $\omega_i \in \mathbb{Z}_e$ is associated with player i. If the ω_i fulfil some assumption, namely that ω_i and $\omega_i - \omega_j$ $(i \neq j)$ are units, then it can be shown that every authorized subset can interpolate the polynomial and hence reconstruct the secret whereas an unauthorized subset cannot.

The aim of this thesis is to combine these three improvements of the Shamir-scheme

- general access structures
- security against active adversaries and
- more general secret-space, namely modules instead of fields

to one scheme. To achieve this, we will unify Karchmer and Wigderson's scheme based on span programs with Desmedt and Frankel's threshold scheme over finite Abelian groups and adapt Cramer, Damgård and Maurer's verifiable scheme to our somewhat more general situation.
After some definitions in chapter 2, we will, motivated by [10], show in chapter 3 that for every R-module E with a finite number of elements, where R is a commutative ring with 1, there exists a threshold scheme over E. Note that every finite Abelian group can be seen as a module over the ring $\mathbb{Z}$ or $\mathbb{Z}_e$, where e is the exponent of the group. In chapter 4, we will introduce *extended* span programs. An extended span program is in fact a span program over a ring R instead of over a field F where, further, the condition that the first unit vector $\mathbf{e}_1 = (1, 0, \ldots, 0) \in F^e$ does not lie in the span of the

vectors of the players in A if A is unauthorized is replaced by the somewhat more general condition that there exists a vector which is perpendicular to all vectors of the players in A and whose first coordinate is a unit. Then we show, how such extended span programs can be used to construct secret sharing schemes over modules for general access structures. Motivated by [8], we will investigate in chapter 5 on what can be done if the dealer and/or the shareholders play faulty. We will show that the schemes developed in the chapter before can be made secure against active adversaries. Finally, in chapter 6, we will, as an application, present a robust RSA *function sharing scheme*, which also can be seen as a robust RSA signature sharing scheme. Note that, coming back to our initial example of a company wanting some groups of people to be able to sign checks and contracts in the name of the company, lets say using an RSA signature scheme, it is not sufficient to share the secret key among the people and then to recover it by the group that wants to sign some document, because then, after the first signature, every single person of this group and every person who listened silently to the key reconstruction knows the secret key and is able to sign further checks on its own. Therefore, either a trusted party has to be included which does the key recovering and the signing or the group has to somehow sign the check, using their shares, without first to reconstruct the secret key. This can be done with the RSA function sharing scheme presented in chapter 6, even if some people try to sabotage the signing by not following the protocol.

Chapter 2

Definitions and Notations

Let l be a positive integer and $\mathcal{P} = \{1, \ldots, l\}$, the set of *players*. A monotone collection Γ of subsets of $\mathcal{P}$ is called an *access structure* over $\mathcal{P}$. Monotone means that if $A \in \Gamma$, then every superset $A' \supseteq A$ is in Γ too. If a subset $A \subseteq \mathcal{P}$ is an element of Γ, then A is called *authorized*, otherwise *unauthorized*.

Let K be some set. A *secret sharing scheme* over K for Γ contains two algorithms, the first, the *distribution* algorithm, which will be executed by the *dealer*, takes as input an element s in K and outputs, besides some public information, l *shares* $s_1, \ldots, s_l$, each being privately sent to the corresponding player, such that any authorized subset of players can reconstruct the secret s from their shares and the public information using the second algorithm, the *reconstruction* algorithm, yet an unauthorized subset of players cannot, using any method.

A secret sharing scheme is *perfect*, if an unauthorized subset not only cannot compute the secret from their shares, but, in the information theoretical sense, their shares give away no information about the secret at all. For $i \in \mathcal{P}$ let S_i be the set where the i-th share s_i lies in. If the S_i are all of the same size as K, then the secret sharing scheme is called *ideal*.

We will only consider secret sharing schemes where K and the S_i carry the structure of a module over some common commutative ring R with 1. Such a secret sharing scheme is called *homomorphic*, if the following property is satisfied. If s_i is player i's share of s, s'_i is i's share of s' and $\lambda \in R$, then

$s_i + s'_i$ and λs_i is i's share of $s + s'$ and λs, respectively.

For $\mathcal{K}$ an infinite subset of $\mathbb{N}$ and $\kappa \in \mathcal{K}$ let E_κ be a finite module [1] over a commutative ring R_κ with 1, not necessarily finite. We say that the family of modules $(E_\kappa)_{\kappa\in\mathcal{K}}$ is *efficient* if addition, subtraction and scalar multiplication in E_κ and the ring operations in R_κ may be computed in polynomial time in $\log\kappa$ and generating uniform random elements of E_κ can be performed in probabilistic polynomial time in $\log\kappa$. Let $\mathcal{L}$ be some infinite subset of $\mathbb{N}$, and for $l \in \mathcal{L}$ let $\mathcal{P}_l$ be the set of players $\mathcal{P}_l = \{1, \ldots, l\}$ and Γ_l an access structure over $\mathcal{P}_l$. Then we call a family of secret sharing schemes, indexed by $l \in \mathcal{L}$ and $\kappa \in \mathcal{K}$, over the modules $(E_\kappa)_{\kappa\in\mathcal{K}}$ for the access structures $(\Gamma_l)_{l\in\mathcal{L}}$ *efficient*, if the distribution and the reconstruction algorithms of the schemes run in polynomial time in $\max(l, \log\kappa)$. From now on we say a module E over a ring R instead of a family of modules $(E_\kappa)_{\kappa\in\mathcal{K}}$ over the family of rings $(R_\kappa)_{\kappa\in\mathcal{K}}$, we say an access structure Γ instead of a family of access structures $(\Gamma_l)_{l\in\mathcal{L}}$ and we say a secret sharing scheme instead of a family of schemes, indexed by l and κ.

[1] According to [15], a *finite* module is a module which has a finite number of generators. In our definition, a finite module is one with a finite number of elements.

Chapter 3

Threshold Schemes

In this chapter we look at a special class of access structures, the so called *threshold access structures*, which are of the form $\Gamma = \{A \subseteq \mathcal{P} \mid |A| \geq t\}$, where $\mathcal{P} = \{1, \ldots, l\}$ is the set of players and $t, 1 \leq t \leq l$, the *threshold*. A secret sharing scheme for such an access structure is called a (t, l)-*threshold scheme*. In [21] Shamir presented a threshold scheme over finite fields and in [10] Desmedt and Frankel generalized this Shamir-scheme to share a secret from a finite Abelian group. For this, they looked at the group as a module over $\mathbb{Z}$ or $\mathbb{Z}_e$, where e is the exponent of the group. In this chapter we present a threshold-scheme over finite R-modules, where R is a commutative ring with 1. We will first show that such a scheme exists under some assumption on R and then that this assumption can always be achieved.

During the whole chapter let E be an efficient finite module over R, where R is a commutative, not necessarily finite ring with 1.

3.1 A Generalized Shamir-Scheme

Let $\Gamma = \{A \subseteq \mathcal{P} \mid |A| \geq t\}$ be a threshold access structure with threshold t over a set $\mathcal{P} = \{1, \ldots, l\}$ of l players. We first assume that there exist l distinct elements $\omega_i \in R$ such that $\omega_i \in R^*$, $i = 1 \ldots, l$, and $\omega_i - \omega_j \in R^*$, $i \neq j$, where R^* denotes the group of units of R. We will call a ring l-*good* if it fulfils this condition.

The following secret sharing scheme over the R-module E is a generalization of the Shamir-scheme, in the sense that if E is a finite field seen as a module over itself, then this scheme coincides with the Shamir-scheme.

Scheme 1

Distribution phase. Let $\omega_1, \ldots, \omega_l \in R$ satisfy the conditions above. The dealer associates ω_i with player i and makes it public. Further, he chooses elements $y_1, \ldots, y_{t-1} \in E$ at random and sets $f(X) = s + y_1 X + \cdots + y_{t-1} X^{t-1}$, where $s \in E$ is the secret the dealer wants to share, and for all players $i \in \mathcal{P}$ he calculates the share $s_i = f(\omega_i)$ and sends it privately to player i.

Reconstruction phase. Let $A \in \Gamma$ with $|A| = t$. Then the secret can be computed as $s = \sum_{i \in A} \eta_{i,A} s_i$, where

$$\eta_{i,A} = \prod_{j \in A, j \neq i} \frac{0 - \omega_j}{\omega_i - \omega_j} \in R$$

Note that the $\eta_{i,A}$ exist as we assumed that $\omega_i - \omega_j \in R^*$. The assumption that also $\omega_i \in R^*$ will be needed for the proof of the perfectness of this scheme. Further, we will need the following lemmas from elementary ring theory, see e. g. [15, pp. 516, 518].

Lemma 3.1 *Let V be a Vandermonde matrix over some commutative ring R, i. e. $V = (\lambda_i^j)_{i,j=0,\ldots,m}$ for some distinct $\lambda_i \in R$, $\lambda_i \neq 0$. Then the determinant of V is*

$$\det(V) = \prod_{i>j} (\lambda_i - \lambda_j)$$

If we define $0^0 = 1$, then lemma 3.1 holds also when one of the λ_i is equal to zero.

Lemma 3.2 *A square matrix over some commutative ring R with 1 is invertible if and only if the determinant of the matrix is a unit in R.*

Proposition 3.3 *Scheme 1 is an efficient, perfect, ideal and homomorphic (t, l)-threshold scheme.*

Proof. We first proof the correctness and then the perfectness, the other claims should be clear. Correctness: Let $A \in \Gamma$. We assume wlog that A is minimal, i. e. $|A| = t$. Consider the polynomial

$$p_A(X) = \sum_{i \in A} (p_{i,A}(X) \cdot s_i)$$

with coefficients in E, where

$$p_{i,A}(X) = \prod_{j \in A, j \neq i} \frac{(X - \omega_j)}{(\omega_i - \omega_j)} \in R[X].$$

¿From the construction of the $p_{i,A}(X)$ it follows that $p_A(\omega_i) = s_i = f(\omega_i)$ for all $i \in A$. We claim that this implies $p_A(X) = f(X)$. ¿From this the correctness follows immediately, namely

$$s = f(0) = p(0) = \sum_{i \in A} p_{i,A}(0) s_i = \sum_{i \in A} \eta_{i,A} s_i$$

To prove that $p_A(X) = f(X)$, we look at the linear equation system

$$f_0 + f_1 \omega_i + \cdots + f_{t-1} \omega_i^{t-1} = s_i, i \in A$$

in the indeterminate f_j, $j = 0, \ldots, t-1$. The corresponding matrix is the Vandermonde matrix $V = (\omega_i^j)_{i \in A, j = 0, \ldots, t-1}$. According to lemma 3.1 the determinant of V is $\det V = \prod_{i > j} (\omega_i - \omega_j)$, which is a unit since $\omega_i - \omega_j$ is a unit. Thus V is, according to lemma 3.2, invertible which induces that the solution of the equation system is unique. This proves that $p_A(X) = f(X)$.
Perfectness: Let $A \notin \Gamma$. We assume wlog that A is maximal, i. e. $|A| = t - 1$. We will show that A's share s_i, $i \in A$, do not reveal any information about the secret, i. e. every $s' \in E$ is equally possible to be the secret. We will do this by showing that for every $s' \in E$ there exists exactly one polynomial $f'(X)$ with coefficients in E such that

$$\begin{aligned} s' &= f'(0) \\ s_i &= f'(\omega_i) \qquad i \in A \end{aligned}$$

For this look at the linear equation system

$$\begin{aligned} s' &= f'_0 \\ s_i &= f'_0 + f'_1 \omega_i + \cdots + f'_{t-1} \omega_i^{t-1} \qquad i \in A \end{aligned}$$

in the indeterminate $f'_j, j = 0, \ldots, t-1$. The determinant of the corresponding matrix is according to lemma 3.2 $\prod_{i \in A} \omega_i \prod_{i,j \in A, i>j}(\omega_i - \omega_j)$, a unit, and therefore there exists exactly one solution to the above equation system what we had to prove.

$\square$

Corollary 3.4 *If A is an unauthorized subset, then $\mathbf{s}_A = (s_i)_{i \in A}$, the superposition of the shares s_i with $i \in A$, is uniformly distributed in $E^{|A|}$.*

3.2 Extension of the Ring

We have seen that if the ring R is l-good, then there exists a (t, l)-threshold scheme over the R-module E for any threshold $t \leq l$. But of course, it might happen that the ring R is not l-good, i. e. there are not enough elements ω_i in R such that ω_i and $\omega_i - \omega_j$ are units as requested in scheme 1. In the Shamir-scheme (over a finite field F) the problem of F being too small can be solved by extending F to a larger field $\bar{F}$ and sharing the secret over this field extension $\bar{F}$.
When the ring R is not l-good we can do something similar. We extend the ring R to a ring $\bar{R}$ which is l-good. Simultaneously we have to extend the module E to a module $\bar{E}$ over the ring $\bar{R}$.
In this section we will present a method to construct such an extension ring $\bar{R}$ and in the next section we will show how the R-module E can be extended to a module $\bar{E}$ over $\bar{R}$ and how this gives rise to a secret sharing scheme over E.

For the first part of this section we will not only assume that R is commutative and contains the 1, but, further, that R is Noetherian, has zero-divisors and every prime ideal is a maximal ideal. It is easy to see that, for instance, all finite rings—commutative and containing the 1— which are no fields fulfil these further assumptions. We will look at the general case in the second part of this section, but we already stress at this stage that the construction of the appropriate ring extension in the general case is less efficient than in this special case we are looking at now.

First, we have a closer look at the structure of R.

Proposition 3.5 *If R is a commutative Noetherian ring with 1 and zero-divisors such that every prime ideal is maximal, then R is of the form*

$$R \cong R/\mathfrak{m}_1^{e_1} \times \ldots \times R/\mathfrak{m}_n^{e_n} \tag{3.1}$$

where the $\mathfrak{m}_i$ are the maximal ideals of R and the e_i positive integers.

Wlog we can assume that $|R/\mathfrak{m}_1| \leq \ldots \leq |R/\mathfrak{m}_n|$ [1].
For the proof of this proposition we need the following lemmas.

Lemma 3.6 *In a commutative Noetherian ring R, each non-zero ideal contains a product of prime ideals.*

Proof. Assume that there exists an ideal $\mathfrak{a} \neq \{0\}$ that does not contain a product of prime ideals. Because R is Noetherian we may wlog say that $\mathfrak{a}$ is maximal with this property. Note that $\mathfrak{a}$ cannot be a prime ideal. So there exist $b, c \in R$ such that $bc \in \mathfrak{a}$ but $b, c \notin \mathfrak{a}$. The ideals $\mathfrak{b} = <\mathfrak{a}, b>$ and $\mathfrak{c} = <\mathfrak{a}, c>$ both contain $\mathfrak{a}$ strictly, therefore $\mathfrak{b}$ and $\mathfrak{c}$ both contain a product of prime ideals, say $\mathfrak{b} \supseteq \prod \mathfrak{p}_i^{\lambda_i}$ and $\mathfrak{c} \supseteq \prod \mathfrak{q}_j^{\mu_j}$. But then $\mathfrak{a} \supseteq \mathfrak{b}\mathfrak{c} \supseteq \prod \mathfrak{p}_i^{\lambda_i} \prod \mathfrak{q}_j^{\mu_j}$, which is a contradiction.

□

Note that if $\mathfrak{a}$ and $\mathfrak{b}$ are ideals in R, then $\mathfrak{a}+\mathfrak{b}$, $\mathfrak{a}\cap\mathfrak{b}$ and $\mathfrak{a}\mathfrak{b}$ denote the ideals

$$\begin{aligned} \mathfrak{a}+\mathfrak{b} &= \{a+b \mid a \in \mathfrak{a}, b \in \mathfrak{b}\} \\ \mathfrak{a}\cap\mathfrak{b} &= \{c \mid c \in \mathfrak{a}, c \in \mathfrak{b}\} \\ \mathfrak{a}\mathfrak{b} &= <\{ab \mid a \in \mathfrak{a}, b \in \mathfrak{b}\}> \end{aligned}$$

Lemma 3.7 *For any commutative ring R with 1, the following holds. If $\mathfrak{a}$ and $\mathfrak{b}$ are ideals such that $\mathfrak{a}+\mathfrak{b} = R$, i. e. $\mathfrak{a}$ and $\mathfrak{b}$ are coprime, then $\mathfrak{a}\mathfrak{b} = \mathfrak{a}\cap\mathfrak{b}$ and $\mathfrak{a}^r + \mathfrak{b}^s = R$ for all $r, s \in \mathbb{N}$.*

Proof. The first claim is well known from elementary ring theory, see e. g. [15, p. 87]. For the second claim it is sufficient to show that $\mathfrak{a}^r + \mathfrak{b} = R$. By induction, let $a_{r-1} \in \mathfrak{a}^{r-1}$ and $b' \in \mathfrak{b}$ such that $a_{r-1} + b' = 1$. Further, let $a \in \mathfrak{a}$ and $b \in \mathfrak{b}$ such that $a + b = 1$. Then $1 = (a+b)(a_{r-1}+b') = aa_{r-1} + (ab' + a_{r-1}b + bb') \in \mathfrak{a}^r + \mathfrak{b}$.

□

[1] If some of the $R/\mathfrak{m}_k$ are infinite, then we can assume that $|R/\mathfrak{m}_1| \leq \ldots \leq |R/\mathfrak{m}_{k_0}| < \infty$ and $|R/\mathfrak{m}_{k_0+1}|, \ldots, |R/\mathfrak{m}_n| = \infty$.

Lemma 3.8 *If a commutative ring R with 1 is of the form (3.1) for some maximal ideals $\mathfrak{m}_k$, then $\prod \mathfrak{m}_k^{e_k} = \{0\}$ and*

$$\mathfrak{m} \subseteq R \textit{ maximal } \Longrightarrow \exists\, k_0 : \mathfrak{m} = \mathfrak{m}_{k_0}$$

i. e. the $\mathfrak{m}_k$ are all the maximal ideals of R.

Proof. Using Chinese Reminder Theorem and the above lemma, it follows from $R \cong R/\mathfrak{m}_1^{e_1} \times \ldots \times R/\mathfrak{m}_n^{e_n}$ that $\{0\} = \bigcap \mathfrak{m}_k^{e_k} = \prod \mathfrak{m}_k^{e_k}$. Thus for every maximal ideal $\mathfrak{m}$ there is $\prod \mathfrak{m}_k^{e_k} \subseteq \mathfrak{m}$. But as $\mathfrak{m}$ is prime, it follows that $\mathfrak{m}_{k_0} \subseteq \mathfrak{m}$ for a k_0 and thus, as $\mathfrak{m}_{k_0}$ is maximal, $\mathfrak{m}_{k_0} = \mathfrak{m}$.

□

Proof of proposition 3.5. Let $0 \neq a, b \in R$ such that $ab = 0$. Lemma 3.6 guarantees that the ideals $\mathfrak{a} = <a>$ and $\mathfrak{b} = <b>$ both contain a product of prime ideals, i. e. $\mathfrak{a} \supseteq \prod \mathfrak{p}_i^{\lambda_i}$ and $\mathfrak{b} \supseteq \prod \mathfrak{q}_j^{\mu_j}$ for $\mathfrak{p}_i$ and $\mathfrak{q}_j$ prime ideals. Hence, $\{0\} = \mathfrak{a}\mathfrak{b} \supseteq \prod \mathfrak{p}_i^{\lambda_i} \prod \mathfrak{q}_j^{\mu_j}$. As we assumed that every prime ideal is maximal, this means $\{0\} = \prod \mathfrak{m}_k^{e_k}$ for some maximal and pairwise different ideals $\mathfrak{m}_k$ and positive integers e_k. ¿From lemma 3.7 and the fact that two different prime ideals are always coprime it follows that the $\mathfrak{m}_k^{e_k}$ are pairwise coprime and $\{0\} = \bigcap \mathfrak{m}_k^{e_k}$, and thus Chinese Reminder Theorem implies that $R/\mathfrak{m}_1^{e_1} \times \cdots \times R/\mathfrak{m}_n^{e_n} \cong R/\bigcap \mathfrak{m}_k^{e_k} = R/\{0\} \cong R$. Finally, lemma 3.8 guarantees that the $\mathfrak{m}_k$ are all the maximal ideals of R.

□

The next thing we have to do is to characterize the units in R.

Lemma 3.9 *If R is a commutative Noetherian ring with 1 and $\epsilon \in R$, then ϵ is a unit if and only if $\epsilon \notin \mathfrak{m}$ for all maximal ideals $\mathfrak{m} \subseteq R$.*

Proof. If $\epsilon \in R^*$, then from $\epsilon \in \mathfrak{m}$ it would follow that $\mathfrak{m} = R$, which contradicts the maximality, thus $\epsilon \notin \mathfrak{m}$. Let now $\epsilon \notin \mathfrak{m}$ for all maximal ideals $\mathfrak{m} \subseteq R$. Consider the ideal generated by ϵ and suppose $<\epsilon> \neq R$. Then, as every ideal not equal to R is contained in some maximal ideal, see e. g. [15, p. 93], there exists a maximal ideal $\mathfrak{m}' \subseteq R$ such that $<\epsilon> \subseteq \mathfrak{m}'$. But by $\epsilon \notin \mathfrak{m}$ for all maximal ideals $\mathfrak{m} \subseteq R$, this is a contradiction. Hence, $<\epsilon> = R$ and therefore $\epsilon \in R^*$.

□

Now we are ready to give a sufficient and necessary condition for R to be l-good.

Theorem 1 *The ring R with decomposition (3.1) is l-good if and only if*

$$l \leq |R/\mathfrak{m}_1| - 1$$

Note that $\mathfrak{m}_1$ is the largest of the $\mathfrak{m}_k$ in the sense that $|R/\mathfrak{m}_1|$ is minimal.

Proof. Is $l \leq |R/\mathfrak{m}_1| - 1$, then there exist $\omega_i^{(k)} \in R$, $i \in \mathcal{P}$, such that $\omega_i^{(k)} \not\equiv 0$ and $\omega_i^{(k)} - \omega_j^{(k)} \not\equiv 0 \pmod{\mathfrak{m}_k}$, $i \neq j$, for $k = 1, \ldots, n$. As $R \to R/\mathfrak{m}_1 \times \cdots \times R/\mathfrak{m}_n$ is surjective, which follows from (3.1), there exist $\omega_i \in R$ such that $\omega_i \equiv \omega_i^{(k)} \pmod{\mathfrak{m}_k}$ for all k, $i \in \mathcal{P}$. It follows from the construction and from proposition 3.9 that the ω_i fulfil $\omega_i, \omega_i - \omega_j \in R^*$ for all $i \neq j$. But is $l > |R/\mathfrak{m}_1| - 1$, then for every choice of $\omega_1, \ldots, \omega_l \in R$ it must $\omega_i \equiv 0 \pmod{\mathfrak{m}_1}$ for at least one i or $\omega_i \equiv \omega_j \pmod{\mathfrak{m}_1}$ for at least one i and j, $i \neq j$. But this induces according to proposition 3.9 that ω_i or $\omega_i - \omega_j$, respectively, is not a unit.

□

Now we are so far that we know when the ring R is not l-good. We will in the following show how such a ring can be extended to a ring $\bar{R}$ which is l-good.
For this let m be a positive integer such that

$$l \leq |R/\mathfrak{m}_1|^m - 1$$

Choose a monic polynomial $f(X) = f_0 + f_1 X + \cdots + f_{m-1} X^{m-1} + X^m \in R[X]$ such that for every $k = 1, \ldots, n$ $f(X)$ is irreducible modulo $\mathfrak{m}_k$ and therefore $(R/\mathfrak{m}_k)[X]/{<}f(X){>}_{(R/\mathfrak{m}_k)[X]}$ is a field . This can be done by choosing irreducible monic polynomials of degree m in $(R/\mathfrak{m}_k)[X]$ and using Chinese Reminder Theorem on the coefficients to compute $f(X)$ as requested. $\bar{R} = R[X]/{<}f(X){>}$ is a ring extension of R and by putting $\bar{\mathfrak{m}}_k = {<}\mathfrak{m}_k{>}_{\bar{R}}$ we get

Proposition 3.10 *The $\bar{\mathfrak{m}}_k$ are all the maximal ideals in $\bar{R}$ and*

$$\bar{R} = \bar{R}/\bar{\mathfrak{m}}_1^{e_1} \times \ldots \times \bar{R}/\bar{\mathfrak{m}}_n^{e_n}$$

Further, $|\bar{R}/\bar{\mathfrak{m}}_k| = |R/\mathfrak{m}_k|^m$, $k = 1, \ldots, n$, *especially* $|\bar{R}/\bar{\mathfrak{m}}_1| \leq \ldots \leq |\bar{R}/\bar{\mathfrak{m}}_n|$.

For the proof we need the following lemmas from elementary ring theory. The former is also known as the third ring isomorphism theorem, see e. g. [16, p. 134].

Lemma 3.11 *If $\mathfrak{a} \subseteq \mathfrak{b}$ are two ideals in R, then $R/\mathfrak{b} \cong (R/\mathfrak{a})/(\mathfrak{b}/\mathfrak{a})$.*

Lemma 3.12 *If $\mathfrak{a}, \mathfrak{b} \subseteq R$ are two ideals, then $<\mathfrak{a}, \mathfrak{b}>_R/\mathfrak{b} = <\mathfrak{a}>_{R/\mathfrak{b}}$ in $R/\mathfrak{b}$.*

Proof of proposition 3.10. First we show the maximality of the $\mathfrak{m}_k$. Let $\mathfrak{m}$ be one of the $\mathfrak{m}_k$ and $\bar{\mathfrak{m}}$ the corresponding $\bar{\mathfrak{m}}_k$ and put $\mathfrak{M} = <\mathfrak{m}>_{R[X]}$. Note that $R[X]/\mathfrak{M} \cong (R/\mathfrak{m})[X]$. For the rest of the proof, $<\cdot>$ stands for $<\cdot>_{R[X]}$. Because

$$\begin{aligned}
\bar{R}/\bar{\mathfrak{m}} &= (R[X]/<f(X)>)/<\mathfrak{m}>_{\bar{R}} \\
&\overset{\text{Lemma 3.12}}{=} (R[X]/<f(X)>)/(<\mathfrak{m}, f(X)>/<f(X)>) \\
&\overset{\text{Lemma 3.11}}{\cong} R[X]/<\mathfrak{M}, f(X)> \\
&\overset{\text{Lemma 3.11}}{\cong} (R[X]/\mathfrak{M})/(<\mathfrak{M}, f(X)>/\mathfrak{M}) \\
&\overset{\text{Lemma 3.12}}{=} (R[X]/\mathfrak{M})/<f(X)>_{R[X]/\mathfrak{M}} \\
&\cong (R/\mathfrak{m})[X]/<f(X)>_{(R/\mathfrak{m})[X]}
\end{aligned}$$

and the latter is a field, $\bar{R}/\bar{\mathfrak{m}}$ is a field and therefore $\bar{\mathfrak{m}}$ is maximal in $\bar{R}$. Further, it follows that $|\bar{R}/\bar{\mathfrak{m}}| = |(R/\mathfrak{m})[X]/<f(X)>_{(R/\mathfrak{m})[X]}| = |R/\mathfrak{m}|^m$. Note that for $\mathfrak{M}_i = <\mathfrak{m}_i>$

$$\begin{aligned}
\prod <\mathfrak{M}_i, f(X)>^{e_i} &\subseteq <\prod \mathfrak{M}_i^{e_i}, f(X)> \\
&\subseteq <<\underbrace{\prod \mathfrak{m}_i^{e_i}}_{=\{0\}}>, f(X)> \\
&= <f(X)>
\end{aligned}$$

so therefore

$$\begin{aligned}
\prod \bar{\mathfrak{m}}_i^{e_i} &\overset{\text{Lemma 3.12}}{=} \prod (<\mathfrak{M}_i, f(X)>/<f(X)>)^{e_i} \\
&= \underbrace{\prod <\mathfrak{M}_i, f(X)>^{e_i}}_{\subseteq <f(X)>}/<f(X)> \\
&= \{0\}
\end{aligned}$$

Analogue to the proof of proposition 3.5 it now follows that $\bar{R}$ can be written as $\bar{R} = \bar{R}/\bar{\mathfrak{m}}_1^{e_1} \times \ldots \times \bar{R}/\bar{\mathfrak{m}}_n^{e_n}$. Finally, lemma 3.8 guarantees that the $\bar{\mathfrak{m}}_k$ are all the maximal ideals of $\bar{R}$.

$\square$

Now we know that the cardinality of $\bar{R}/\bar{\mathfrak{m}}_1$ is $|\bar{R}/\bar{\mathfrak{m}}_1| = |R/\mathfrak{m}_1|^m$ and therefore $l \leq |\bar{R}/\bar{\mathfrak{m}}_1| - 1$, which means that $\bar{R}$ is l-good. This proves the following

Theorem 2 *If R is a commutative Noetherian ring with* 1 *and zero-divisors such that every prime ideal is maximal, or R is in fact a field, then there exists an extension ring $\bar{R} = R[X]/{<}f(X){>}$ of R which is l-good. Further,* $\deg f(X) = O(\log l)$.

Corollary 3.13 *If R has characteristic $n \neq 0$, i. e. $n \cdot 1 = \overbrace{1 + \cdots + 1}^{n} = 0$ in R and $n > 0$ is minimal with this property, then there exists a ring extension* $\bar{R} = R[X]/{<}f(X){>}$, $\deg f(X) = O(\log l)$, *which is l-good.*

Proof. R contains $\mathbb{Z}_n$ as a subring [2]. According to theorem 2 there exists a ring extension $\mathbb{Z}_n[X]/{<}f(X){>}$ of $\mathbb{Z}_n$ which is l-good, therefore the corresponding ring extension $\bar{R} = R[X]/{<}f(X){>}$ of R has to be l-good as well.

$\square$

Now we will show that such a ring extension which is l-good can be found for any commutative ring R with 1, even though the construction will be less efficient. For this, we will first describe how the ring $\mathbb{Z}$ can be extended to a ring which is l-good, this was presented in [10], and then we will show that this construction can be applied to any commutative ring with 1.
Of course, the ring $\mathbb{Z}$ is not l-good for any $l > 1$. To find an appropriate extension ring, we choose a monic irreducible polynomial $f(X) \in \mathbb{Z}[X]$. Then, $\mathbb{Z}[X]/{<}f(X){>}$ is an extension of $\mathbb{Z}$. But how do we have to chose $f(X)$ such that the extension $\mathbb{Z}[X]/{<}f(X){>}$ is l-good? We have to be aware that in general we cannot use the (extended) Euclidean algorithm in $\mathbb{Z}[X]$. Therefore, $\gcd(f(X), g(X)) = 1$ does not necessarily imply that $g(X)$ is invertible modulo $f(X)$. There exists a special class of polynomials $f(X)$ such that the above question can easily be answered.
Let q be a prime greater then l and $f(X)$ the cyclotomic polynomial

$$f(X) = \frac{X^q - 1}{X - 1} = 1 + X + \cdots + X^{q-1} \in \mathbb{Z}[X]$$

[2] To be precise, R contains a subring which is isomorphic to $\mathbb{Z}_n$

Bertrand's Postulate [17, p. 243] guarantees that q can be chosen smaller as $2l$. It is known from elementary polynomial ring theory that $f(X)$ is irreducible [15, pp. 279, 280]. The following proposition guarantees that the extension ring $\mathbb{Z}[X]/\!<\!f(X)\!>$ is l-good. Note that $\mathbb{Z}[X]/\!<\!f(X)\!> \cong \mathbb{Z}[u]$, where u is a zero of $f(X)$ in some extension ring of $\mathbb{Z}$.

Proposition 3.14 *For $i \in \mathcal{P}$ let $\omega_i = \sum_{k=0}^{i-1} u^k \in \mathbb{Z}[u]$. Then ω_i and $\omega_i - \omega_j$ are units in $\mathbb{Z}[u]$ for all $i \neq j$.*

For the proof we need the following lemma.

Lemma 3.15 *Let $f(X) = 1 + X + \cdots + X^m$ and $g(X) = 1 + X + \cdots + X^n$ in $\mathbb{Z}[X]$. Then there exist polynomials $r(X), s(X) \in \mathbb{Z}[X]$ such that*

$$s(X)f(X) + t(X)g(X) = \gcd(f(X), g(X))$$

Following the Euclidean algorithm, it is quite easy to see that even though in general it does not work for polynomials in $\mathbb{Z}[X]$, it does work for $f(X)$ and $g(X)$ as in the lemma. Therefore, such $s(X)$ and $t(X)$ exist. Still, we give a formal proof.

Proof. For simplification, we will omit the argument X during this proof. Induction on $m + n$. If $m + n = 0$ and therefore $f = g = 1$, then $s = 0$ and $t = 1$ fulfills $sf + tg = 1 = \gcd(f, g)$. If $m + n > 0$ and wlog $m > n$ (the case $m = n$ is trivial), then we put $h = f - X^{m-n}g = 1 + X + \cdots + X^{m-n-1}$. Then, $\gcd(f, g) = \gcd(g, h)$ and, as $\deg h < m$, by induction there exist $r, s \in \mathbb{Z}[X]$ such that $rg + sh = \gcd(g, h)$. For $t = r - X^{m-n}s$ we then have

$$\begin{aligned} sf + tg &= s(h + X^{m-n}g) + (r - X^{m-n}s)g \\ &= sh + rg \\ &= \gcd(g, h) \\ &= \gcd(f, g) \end{aligned}$$

which is what we had to prove.

□

Proof of proposition 3.14. As the polynomial $f(X)$ is irreducible and has degree greater than $\deg(\sum_{k=0}^{i-1} X^k)$, it must $\gcd(f(X), \sum_{k=0}^{i-1} X^k) = 1$. The above lemma 3.15 implies that $\sum_{k=0}^{i-1} X^k$ is invertible modulo $f(X)$ and thus $\omega_i = \sum_{k=0}^{i-1} u^k$ is a unit in $\mathbb{Z}[u]$. Further, note that u^k, $0 \leq k \leq q-1$, is invertible, namely $(u^k)^{-1} = u^{q-k}$, and therefore $\omega_i - \omega_j = \sum_{k=j}^{i-1} u^k = u^j \omega_{i-j}$ is a unit.

□

Finally, we show that this construction we applied to $\mathbb{Z}$ can be applied to any commutative ring with 1.

Theorem 3 *Let R be a commutative ring with 1 and $l \in \mathbb{N}$. Then there exists a ring extension $\bar{R} = R[X]/<f(X)>$ of R which is l-good. Further,* $\deg f(X) = O(l)$.

Proof. If R has characteristic 0, i. e. $n \cdot 1 \neq 0$ in R for all $n > 0$, then R contains $\mathbb{Z}$ as a subring. As $\mathbb{Z}[X]/<f(X)>$ constructed as above is l-good, $\bar{R} = R[X]/<f(X)>$, which contains $\mathbb{Z}[X]/<f(X)>$, must be l-good as well. If the characteristic of R is a positive integer n, then the claim follows from corollary 3.13.

□

We have to be aware that even theorem 3 is more general than theorem 2, the ring extension constructed to prove the former is much smaller than the one constructed to prove the latter. In the former, the degree of the polynomial $f(X)$ is about $\log l$, whereas in the latter, the degree of $f(X)$ is about l.

3.3 Extension of the Module

As mentioned in the beginning of the previous section, we not only have to extend the ring R to a ring $\bar{R}$ which is l-good, we also have to extend the R-module E to a module $\bar{E}$ over $\bar{R}$ to be able to apply scheme 1.
This can be done the following way. Consider the R-module $\bar{R} \otimes E$. By

$$\begin{array}{rcl} \bar{R} \times (\bar{R} \otimes E) & \longrightarrow & \bar{R} \otimes E \\ (\bar{\lambda}, \sum \bar{\mu}_i \otimes x_i) & \longmapsto & \sum \bar{\lambda}\bar{\mu}_i \otimes x_i \end{array}$$

we can extend the scalar multiplication over R to a scalar multiplication over the extension ring $\bar{R}$ and thus extend the R-module $\bar{R} \otimes E$ to a module over $\bar{R}$ which we will denote as $\bar{R} \otimes_R E$ (more on this can be found in [15, p. 623]).

By identifying $x \in E$ with $1 \otimes x \in \bar{R} \otimes_R E$, E can in a natural way be seen as a submodule of $\bar{E} = \bar{R} \otimes_R E$.
In [10] Desmedt and Frankel gave a more constructive description of $\bar{R} \otimes_R E$ in the special case where E is an Abelian group, seen as a module over $R = \mathbb{Z}_e$. In the following we generalize their construction. For this we look at the product $E^m = E \times \ldots \times E$, where m still denotes the degree of $f(X)$, with the component-wise addition $(x_1, \ldots, x_m) + (y_1, \ldots, y_m) = (x_1 + y_1, \ldots, x_m + y_m)$. We can define a scalar multiplication over $\bar{R} = R[X]/<f(X)>$ as follows. Let $\mathbf{x} = (x_1, \ldots, x_m)$ be in E^m and $\bar{\lambda} \in \bar{R}$. Note that $\bar{\lambda}$ can be uniquely represented by a polynomial $\lambda_0 + \lambda_1 X + \cdots + \lambda_{m-1} X^{m-1} \in R[X]$. We now define

$$\bar{\lambda}\mathbf{x} = \sum_{i=0}^{m-1} \lambda_i X^i \mathbf{x}$$

where recursively

$$\lambda_i X^i \mathbf{x} = X(\lambda_i X^{i-1} \mathbf{x})$$

and

$$\begin{aligned} \lambda_i(x_1, \ldots, x_m) &= (\lambda_i x_1, \ldots, \lambda_i x_m) \qquad \text{and} \\ X(x_1, \ldots, x_m) &= (0, x_1, \ldots, x_{m-1}) + (-f_0 x_m, -f_1 x_m, \ldots, -f_{m-1} x_m) \end{aligned}$$

This makes E^m to a module over $\bar{R}$. It is easy to see that the function

$$\begin{array}{rcl} \bar{R} \otimes_R E & \longrightarrow & E^m \\ \sum \bar{\lambda}_i \otimes x_i & \longmapsto & \sum \bar{\lambda}_i (x_i, 0, \ldots, 0) \end{array}$$

is an isomorphism.

Now, coming back to what we actually want, namely a secret sharing scheme over the R-module E, a secret $s \in E$ can be shared by sharing $(s, 0, \ldots, 0) \in E^m$ using scheme 1. Note that the scheme we obtain this way is perfect and homomorphic as scheme 1 but it is not ideal as the secret space E is smaller than the share space $\bar{E}$.
Therefore, we have finally proven

Theorem 4 *Let E be a finite module over a commutative ring with 1. Then there exists an efficient, perfect and homomorphic (t, l)-threshold scheme over E for any threshold t, $1 \leq t \leq l$.*

3.4 Application: Secret Sharing over Abelian Groups

Let G be an efficient finite Abelian group, that means G is in fact a whole family $(G_\kappa)_{\kappa\in\mathcal{K}}$ of finite Abelian groups in which the group operation and the inversion may be computed in polynomial time in $\log\kappa$ and generating uniform random elements of G_κ can be performed in probabilistic polynomial time. We assume wlog that G is additive. If e is the exponent of G, i. e. $e = \min\{n \in \mathbb{N} \mid ng = 0 \,\forall\, g \in G\}$, then G can in a natural way be seen as an efficient module over the ring $\mathbb{Z}_e$. So by applying scheme 1 to the $\mathbb{Z}_e$-module G or to the $(\mathbb{Z}_e[X]/<f(X)>)$-module G^m if G is not l-good, which is the case when $p-1 < l$ for the smallest prime factor p of e, we get a perfect, homomorphic (t,l)-threshold scheme over the group G.
Note that this method to share a secret from G only works if, first, the dealer knows the exponent e of G and, second, the shareholders are allowed to know e as well, which is not the case in many cryptographic algorithms (e. g. in RSA [20] $\varphi(n)$ has to be kept secret).

What we want, or need, is a so called *zero-knowledge* secret sharing scheme. This is a scheme such that every unauthorized subset A of players can, with their knowledge before they got their shares, construct a probabilistic, in $\max(l, \log\kappa)$ polynomial time algorithm, which outputs *simulated* shares for the players A and public information which are indistinguishable from those generated in the distribution phase of the scheme for any secret. Such a zero-knowledge secret sharing scheme guarantees that an unauthorized subset not only gets no information about the secret, but they get no new information at all.
A *minimal-knowledge* secret sharing scheme guarantees that every authorized subset gets no new information except the secret (and what follows from this). This is the case if every authorized subset A can, with their knowledge before they got their shares, construct a polynomial time algorithm which takes a possible secret as input and outputs correct shares of this secret for the players A and public information which are indistinguishable from those generated in the distribution phase of the scheme for the same secret.

In [10] Desmedt and Frankel presented a secret sharing scheme over the finite Abelian group G which is zero-knowledge and minimal-knowledge. For this, we have to look at G as a module over $\mathbb{Z}$. By extending G to the module G^m over $\mathbb{Z}[u]$ as described in the previous sections and applying scheme 1 we get a secret sharing scheme over G which is zero-knowledge and minimal-knowledge. For instance, according to corollary 3.4 an unauthorized subset A can simulate their shares and the public information by choosing elements from G^m with uniform probability and computing the deterministic public information $\omega_i = \sum_{k=0}^{i-1} u^k$ for $i \in \mathcal{P}$.

This simulator only works if the players can pick elements from G (or G^m) with uniform probability. This, of course, is not the case with the group $G = \mathbb{Z}_{\varphi(n)}$ in RSA, as the players do not know $\varphi(n)$. It is shown in [9] that the distribution of choosing a random element from $\mathbb{Z}_n$ and choosing a random element from $\mathbb{Z}_{\varphi(n)}$, both elements being represented by an integer in $\{0, 1, \ldots, n-1\}$ and $\{0, 1, \ldots, \varphi(n) - 1\}$, respectively, are *statistical* indistinguishable for large p and q, the two prime factors of n. This means that the number of samples needed to distinguish the two distributions is larger than polynomial in $\min(\log p, \log q)$. Therefore, the players can simulate their shares by choosing random elements from $\mathbb{Z}_n$ instead of $\mathbb{Z}_{\varphi(n)}$. Thus, we have a secret sharing scheme over $\mathbb{Z}_{\varphi(n)}$ which is statistical zero-knowledge.

Chapter 4

Extended Span Programs

In [14] Karchmer and Widgerson introduced *span programs* as a linear algebraic model of computation. They also described how *monotone* span programs can be used to construct secret sharing schemes over fields for general access structures. As we are only interested in *monotone* span programs, we will from now on skip the word monotone and refer to them just as span programs. In this chapter, we will construct secret sharing schemes over modules. For this we will have to extend the definition of Karchmer and Wigderson's span programs [14], which are defined over fields, to span programs over rings, and to stress the difference, we will (sometimes) call the latter *extended* span programs. Further, we will show that the Shamir-scheme discussed in the previous chapter can be expressed in terms of extended span programs and how this can be used to construct span programs.

During the whole chapter, E is an efficient finite module over a commutative ring R with 1, not necessarily finite. Remember that a module is called efficient if all the module and ring operations and choosing random elements can be performed in (probabilistic) polynomial time. Further, we assume that in R linear equation systems can be solved. We will show in the appendix A that this for instance is the case in the rings $\mathbb{Z}$, $\mathbb{Z}_n$, $\mathbb{Z}[X]$ and $\mathbb{Z}[X]/{<}f(X){>}$.

4.1 Definition

Let M be a matrix with entries in R, having d rows and e columns. We will write this as $M \in R^{d \times e}$. We assume that M is labelled in the sense that every row is indexed by some integer $i \in \mathcal{P} = \{1, \ldots, l\}$, where every index i occurs at least once. Finally, let $\mathbf{0} \neq \mathbf{u} \in R^e$. An *extended span program* over R is a triple $(R, M, \mathbf{u})$, defined as above.
By M_i, $1 \leq i \leq l$, we denote the matrix that consists of the rows in M indexed by i. Similarly, if $\emptyset \neq A \subseteq \mathcal{P}$, M_A stands for the matrix consisting of the rows in M indexed with elements $i \in A$. d_i and d_A denote the number of rows of M_i and M_A, respectively. Let Γ be an access structure over $\mathcal{P}$. If for all $A \subseteq \mathcal{P}$

$$\begin{aligned} A \in \Gamma \quad &\Longrightarrow \quad \mathbf{u} \in \operatorname{Im} M_A^t \text{ and} \\ A \notin \Gamma \quad &\Longrightarrow \quad \exists\, \mathbf{a} \in \operatorname{Ker} M_A \subseteq R^e : \langle \mathbf{u}, \mathbf{a} \rangle \in R^* \end{aligned}$$

holds, then the extended span program is said to *compute* Γ.
The following lemma guarantees that $\operatorname{Ker} M_A = (\operatorname{Im} M_A^t)^\perp$. This implies that the second condition is in general stronger than $\mathbf{u} \notin \operatorname{Im} M_A^t$, the corresponding condition of the span program introduced by Karchmer and Wigderson in [14], and equivalent if R is a field.

Lemma 4.1 *Any matrix M with entries in R fulfils* $\operatorname{Ker} M = (\operatorname{Im} M^t)^\perp$.

Note that this claim is a well known fact in the case where R is a field.

Proof. It is easy to verify that $\langle M\mathbf{b}, \mathbf{c} \rangle = \langle \mathbf{b}, M^t\mathbf{c} \rangle$ for all vectors $\mathbf{b}$ and $\mathbf{c}$ with proper dimensions. So let first $\mathbf{b}$ be an element of $\operatorname{Ker} M$, i. e. $M\mathbf{b} = \mathbf{0}$. Then for all $\mathbf{c}$ we have $0 = \langle M\mathbf{b}, \mathbf{c} \rangle = \langle \mathbf{b}, M^t\mathbf{c} \rangle$, hence, $\mathbf{b} \in (\operatorname{Im} M^t)^\perp$. Now let $\mathbf{b} \in (\operatorname{Im} M^t)^\perp$. Then $0 = \langle \mathbf{b}, M^t\mathbf{c} \rangle = \langle M\mathbf{b}, \mathbf{c} \rangle$ for all $\mathbf{c}$, especially for $\mathbf{c} = \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_e$, where $\mathbf{e}_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ [1] with the 1 in the i-th position, thus $M\mathbf{b} = \mathbf{0}$.

$\square$

We call an extended span program $(\bar{R}, M, \mathbf{u})$ *efficient* over R, if $\bar{R}$ is of the form $\bar{R} = R[X]/{<}f(X){>}$ and the number of rows and columns of M, the length of the entries of M and the degree of $f(X)$ are bounded by a polynomial in $\max(l, \log \kappa)$.

[1] Even though we write the vectors as row vectors, they are normally to be seen as column vectors.

4.2 Sharing a Secret from a Ring

This extension from a span program over a field to a span program over a ring gives an extension from Karchmer and Widgerson's secret sharing schemes over finite fields to schemes over finite rings. So let R be a finite commutative ring with 1, Γ an access structure over a set $\mathcal{P}$ of l players and $(R, M, \mathbf{u})$, $M \in R^{d \times e}$ and $\mathbf{u} \in R^e$, a span program that computes Γ. If R is a field and therefore R^e a vector space, we can, just by changing the basis, achieve that $\mathbf{u} = \mathbf{e}_1 = (1, 0, \ldots, 0)$. In the general case where R is not necessarily a field we can also assume wlog that $\mathbf{u} = \mathbf{e}_1$, as follows from

Proposition 4.2 *If $(R, M, \mathbf{u})$, $M \in R^{d \times e}$ and $\mathbf{u} \in R^e$, is an extended span program computing Γ, then there exists an extended span program $(R, M', \mathbf{e}_1)$, $M' \in R^{d' \times e'}$ and $\mathbf{e}_1 \in R^{e'}$, computing Γ with d' and e' bounded by $d' \leq d + l$ and $e' \leq e + 1$.*

Proof. We proof proposition 4.2 in two steps. In step one we show that $(R, M', \mathbf{e}_1)$ exists under some assumption and in step two we show that this assumption can always be achieved. Step one: We assume that one coordinate of $\mathbf{u} = (\lambda_1, \ldots, \lambda_e)^t$ is a unit, wlog we say $\lambda_1 = 1$. By choosing the basis $\mathbf{u}, \mathbf{e}_2, \ldots, \mathbf{e}_e$ of R^e, $(R, M, \mathbf{u})$ transforms to $(R, M', \mathbf{e}_1)$, $M' \in R^{d \times e}$ and $\mathbf{e}_1 \in R^e$. Note that if λ_1 is not a unit, then $\mathbf{u}, \mathbf{e}_2, \ldots, \mathbf{e}_e$ is not a basis of R^e. Step two: By adding a $(e+1)$-st column, filled with zeros, and l rows $(0, \ldots, 0, 1)$ of length $e+1$, labelled with i from 1 to l, to the matrix M and by replacing $\mathbf{u} \in R^e$ by $\binom{\mathbf{u}}{1} \in R^{e+1}$, we get a span program computing Γ which fulfils the assumption needed in step one. The bounds on d' and e' follow from the construction in step two.

$\square$

Remember, for the following secret sharing scheme we assumed that the ring R is finite.

Scheme 2

Distribution phase. $(R, M, \mathbf{e}_1)$ is public knowledge. Let $s \in R$ be the secret. The dealer chooses random elements $\beta_2, \ldots, \beta_e \in R$ and sets $\mathbf{b} = (s, \beta_2, \ldots, \beta_e)$. For every player $i \in \mathcal{P}$ he then computes the share $\mathbf{s}_i = M_i \mathbf{b} \in R^{d_i}$ and sends it privately to player $i \in \mathcal{P}$.

> *Reconstruction phase.* Let $A \in \Gamma$, and let $\mathbf{v}_A$ be in R^{d_A} such that $M_A^t \mathbf{v}_A = \mathbf{e}_1$. If $\mathbf{s}_A \in F^{d_A}$ is the superposition of the $\mathbf{s}_i$, $i \in A$, then the secret can be computed as $s = \langle \mathbf{v}_A, \mathbf{s}_A \rangle$.

Note, because linear equation systems can be solved in the ring R, as we assumed, the vector $\mathbf{v}_A$ can be computed by the players in A.

Proposition 4.3 *If the span program $(R, M, \mathbf{e}_1)$ is efficient and computes Γ, then scheme 2 is an efficient, homomorphic and perfect secret sharing scheme for Γ. If $d_i = 1$ for all $i \in \mathcal{P}$, it is ideal.*

Proof. We first proof the correctness and then the perfectness, the other claims should be clear. Correctness: Let $A \in \Gamma$ and $\mathbf{v}_A$ such that $M_A^t \mathbf{v}_A = \mathbf{e}_1$. Then

$$\langle \mathbf{v}_A, \mathbf{s}_A \rangle = \langle \mathbf{v}_A, M_A \mathbf{b} \rangle = \langle M_A^t \mathbf{v}_A, \mathbf{b} \rangle = \langle \mathbf{e}_1, \mathbf{b} \rangle = s$$

Perfectness: Let $A \notin \Gamma$. The solution space of the equation $M_A \mathbf{b}' = \mathbf{s}_A$ is $\mathbf{b} + \operatorname{Ker} M_A$, and as there exists $\mathbf{x} \in \operatorname{Ker} M_A$ whose first coordinate is a unit, $M_A \mathbf{b}' = \mathbf{s}_A$ has the same number of solutions for each possible choice of a secret s' in the first coordinate of $\mathbf{b}'$.

$\square$

In the next section we will extend this secret sharing scheme to a scheme over a module.

4.3 Sharing a Secret from a Module

In this section we will present a secret sharing scheme based on span programs that allows us to share a secret s from the R-module E. The simplest way to do so would be to share the coefficients of s in respect to a fixed basis of E, using scheme 2. But to do so we have to assume that a basis of E exists, which e. g. is not the case if R is infinite, and, further, is known or easy to compute. The scheme we are going to present below is basis free, but, as we will see at the end of this section, equivalent to sharing the coefficients of the secret in respect to a fixed basis in the case where E has a basis.

But before we can present the scheme, we have to introduce some notations. Is $\mathbf{a} = (\lambda_1, \ldots, \lambda_e) \in R^e$ and $\mathbf{x} = (x_1, \ldots, x_e) \in E^e$, then we put

$$\langle \mathbf{a}, \mathbf{x}] = \sum_{i=1}^{e} \lambda_i x_i \in E$$

which is linear in both arguments and $\langle \mathbf{e}_1, \mathbf{x}] = x_1$. We use this asymmetric notation to stress the asymmetry of the two arguments. If $N = (n_{ij}) \in R^{d \times e}$, we define the matrix multiplication with an element $\mathbf{x} = (x_1, \ldots, x_e) \in E^e$ as

$$N\mathbf{x} = (\sum_{j=1}^{e} n_{1j} x_j, \ldots, \sum_{j=1}^{e} n_{dj} x_j) \in E^d$$

This makes N to a linear map $N : E^e \to E^d$. Of course, N also acts on a vector in R^e with the normal matrix multiplication. It is easy to see that $\langle N\mathbf{a}, \mathbf{x}] = \langle \mathbf{a}, N^t \mathbf{x}]$ where in the former expression N is seen as $N : R^e \to R^d$ and in the latter as $N : E^e \to E^d$. Further, if $N \in R^{d \times 1}$, i. e. N is in fact a (column) vector, then $N\mathbf{x} = \langle N, \mathbf{x}]$.
Note that in the case where the module E is in fact R, seen as a module over itself, $\langle \cdot, \cdot]$ is the standard inner product $\langle \cdot, \cdot \rangle$ over R^p and $N : R^e \to R^d$ coincides with $N : E^e \to E^d$.

Now we are ready to present the secret sharing scheme. Let Γ be an access structure over a set $\mathcal{P}$ of l players and $(R, M, \mathbf{u})$, $M \in R^{d \times e}$ and $\mathbf{u} \in R^e$, a span program that computes Γ. In the last section we have seen that we can wlog assume that $\mathbf{u} = \mathbf{e}_1 = (1, 0, \ldots, 0)$. A secret sharing scheme is now constructed as follows.

Scheme 3

Distribution phase. $(R, M, \mathbf{e}_1)$ is public knowledge. Let $s \in E$ be the secret. The dealer chooses random elements $x_2, \ldots, x_e \in E$ and sets $\mathbf{x} = (s, x_2, \ldots, x_e)$. For every player $i \in \mathcal{P}$ he then computes the share $\mathbf{s}_i = M_i \mathbf{x} \in E^{d_i}$ and sends it privately to player $i \in \mathcal{P}$.

Reconstruction phase. Let $A \in \Gamma$, and let $\mathbf{v}_A$ be in R^{d_A} such that $M_A^t \mathbf{v}_A = \mathbf{e}_1$. If $\mathbf{s}_A \in R^{d_A}$ is the superposition of the $\mathbf{s}_i$, $i \in A$, then the secret can be computed as $s = \langle \mathbf{v}_A, \mathbf{s}_A]$

Proposition 4.4 *If the span program $(R, M, \mathbf{e}_1)$ is efficient and computes Γ, then scheme 3 is an efficient, homomorphic and perfect secret sharing scheme for Γ. If $d_i = 1$ for all $i \in \mathcal{P}$, it is ideal.*

Proof. We again only proof the correctness and the perfectness. Correctness: Let $A \in \Gamma$ and $\mathbf{v}_A$ such that $M_A^t \mathbf{v}_A = \mathbf{e}_1$. Then

$$\langle \mathbf{v}_A, \mathbf{s}_A] = \langle \mathbf{v}_A, M_A \mathbf{x}] = \langle M_A^t \mathbf{v}_A, \mathbf{x}] = \langle \mathbf{e}_1, \mathbf{x}] = s$$

Perfectness: Let $A \notin \Gamma$, thus there exists $\mathbf{a} = (\lambda_1, \ldots, \lambda_e) \in \operatorname{Ker} M_A \subseteq R^e$ with $\lambda_1 = 1$. As for every $y \in E$ there exists $\mathbf{y} \in \operatorname{Ker}(M_A)_E \subseteq E^e$ with y in the first coordinate, namely $\mathbf{y} = \mathbf{a} \cdot y = (\lambda_1 y, \ldots, \lambda_e y)$, the equation $M_A \mathbf{x}' = \mathbf{s}_A$ has the same number of solutions for each possible choice of a secret $s' \in E$.

□

Therefore, we have proven the following

Theorem 5 *If the access structure Γ is computed by an efficient extended span program over R, then there exists an efficient, homomorphic and perfect secret sharing scheme over E for Γ.*

Finally, we show that if E has a basis, which implies that R is finite, then scheme 3 is equivalent to sharing all coordinates of the secret in respect to a fixed basis using scheme 2.

Proposition 4.5 *In the case where $E = R^m$ for a finite commutative ring R with 1 and an integer $m \in \mathbb{N}$, scheme 3 is equivalent to sharing the m coordinates of $\mathbf{s} \in R^m$ independently using scheme 2.*

Proof. Let $\mathbf{s} = (s^{(1)}, \ldots, s^{(m)}) \in R^m$ be the secret and $\mathbf{x}_i = (x_i^{(1)}, \ldots, x_i^{(m)}) \in R^m$, $i = 2, \ldots, e$, the random elements chosen by the dealer in the distribution phase. Look at $\mathbf{s}_i$, the share that player $i \in \mathcal{P}$ gets. Is $M_i^{(jk)}$ the entry in the j-th row and k-th column of M_i, then

$$\begin{aligned} \mathbf{s}_i &= M_i \mathbf{x} \\ &= (M_i^{(j1)} \mathbf{s} + \sum_{k=2}^{e} M_i^{(jk)} \mathbf{x}_k)_{j=1,\ldots,d_i} \end{aligned}$$

$$= \quad ((M_i^{(j1)}s^{(n)} + \sum_{k=2}^{e} M_i^{(jk)} x_k^{(n)})_{n=1,\ldots,m})_{j=1,\ldots,d_i}$$

Note that $\mathbf{s}_i$ is a d_i-dimensional vector with entries in R^m. Looking at the coefficients with $n = n_0$ we have

$$(M_i^{(j1)}s^{(n_0)} + \sum_{k=2}^{e} M_i^{(jk)} x_k^{(n_0)})_{j=1,\ldots,d_i} = M_i \mathbf{x}^{(n_0)}$$

where $\mathbf{x}^{(n_0)} = (s^{(n_0)}, x_2^{(n_0)}, \ldots, x_e^{(n_0)})$, that is $s^{(n_0)}$ shared correctly using scheme 2. Note that the $x_k^{(n)}$ are just random elements from R. So the equivalence of the distribution phase is proven. The equivalence of the reconstruction phase can be shown similarly.

□

4.4 Span Programs and the Shamir-Scheme

In this section we will show that the Shamir-scheme presented in chapter 3 can be expressed in terms of span programs. We first note that we can describe scheme 1 in a slightly different way. We can say that the dealer puts $\mathbf{y} = (s, y_1, \ldots, y_{t-1}) \in E^t$, where $y_1, \ldots, y_{t-1} \in E$ are chosen independently at random and $s \in E$ is the secret he wants to share. Then for every player $i \in \mathcal{P}$ he defines the vector $\mathbf{w}_i = (1, \omega_i, \ldots, \omega^{t-1}) \in R^t$, where the $\omega_i \in R$ are distinct elements such that $\omega_i, \omega_i - \omega_j \in R^*$ for all $i \neq j$. Finally, he hands each player his share

$$s_i = \langle \mathbf{w}, \mathbf{y}\,] = s + y_1\omega_i + \cdots + y_{t-1}\omega_i^{t-1}$$

It is now easy to see that scheme 1 is just a special case of scheme 3, namely when M is the matrix whose l rows are the vectors $\mathbf{w}_i$, indexed by i, for i from 1 to l. With this in mind, the following proposition is not surprising.

Proposition 4.6 *If the matrix M is constructed as described above, then the extended span program $(R, M, \mathbf{e}_1)$ computes the threshold access structure $\Gamma = \{A \subseteq \mathcal{P} \mid |A| \geq t\}$.*

Proof. We only have to show that for $|A| = t$ the equation

$$M_A\mathbf{x} = \mathbf{e}_1$$

and for $|A| = t-1$ the system

$$\begin{aligned} \langle \mathbf{e}_1, \mathbf{x} \rangle &= 1 \\ M_A\mathbf{x} &= \mathbf{0} \end{aligned}$$

is solvable. But this follows straight from lemma 3.1 and 3.2 as in the proof of proposition 3.3. Remember that the rows of M are of the form $(1, \omega_i, \ldots, \omega^{t-1})$, where $\omega_i, \omega_i - \omega_j \in R^*$ for $i \neq j$.

□

4.5 Construction of Extended Span Programs

We have seen how we can share a secret for a given access structure Γ over $\mathcal{P} = \{1, \ldots, l\}$, if we have a span program computing Γ. In this section we investigate on how such a span program may be constructed.
Benaloh and Leichter showed in [2] how a monotone circuit, consisting of `AND` and `OR` gates with an arbitrary number of inputs but only one output, that recognizes Γ can be used to build up a secret sharing scheme for Γ. We will show how such a monotone circuit can be used to construct a span program that computes Γ. But we will allow the circuits not only to consist of `AND` and `OR` gates but of arbitrary threshold gates. But still we consider that every gate has only one output wire.
A monotone circuit C with n boolean inputs $x_1, \ldots, x_n$, called input wires, and one output, $y = C(x_1, \ldots, x_n)$, where every input wire x_j *belongs* to a player $i \in \mathcal{P}$, written as $i = \mathrm{bel}(j)$ [2], is called to *recognize* the access structure Γ, if for $\epsilon_1, \ldots, \epsilon_l \in \{0, 1\}$

$$\{i \in \mathcal{P} \mid \epsilon_i = 1\} \in \Gamma \iff C(\epsilon_{\mathrm{bel}(1)}, \ldots, \epsilon_{\mathrm{bel}(n)}) = 1$$

Further, we say that an extended span program $(R, M, \mathbf{e}_1)$ *computes* a circuit C, if $(R, M, \mathbf{e}_1)$ computes the access structure $\bar{\Gamma} = \{A \subseteq \bar{\mathcal{P}} \mid C(A) = 1\}$ over

[2] $\mathrm{bel}(\cdot) : \{1, \ldots, n\} \to \mathcal{P}$ does not have to be injective, several input wires may belong to the same player.

$\bar{\mathcal{P}} = \{1, \ldots, n\}$, where $C(A)$ stands for $C(x_1, \ldots, x_n)$ with $x_j = 1 \Leftrightarrow j \in A$, i. e. for $A \subseteq \bar{\mathcal{P}}$ we have

$$\begin{aligned} C(A) = 1 &\implies \mathbf{e}_1 \in \operatorname{Im} M_A^t \\ C(A) = 0 &\implies \exists\, \mathbf{a} \in \operatorname{Ker} M_A \subseteq R^e : \langle \mathbf{e}_1, \mathbf{a} \rangle \in R^* \end{aligned}$$

Note that if $(R, M, \mathbf{e}_1)$ computes C, then we only have to change the labelling on M to get a span program that computes Γ. Indeed, if we replace every index $j \in \bar{\mathcal{P}}$ of the rows of M by the corresponding index $i = \mathrm{bel}(j)$, then $(R, M, \mathbf{e}_1)$ computes Γ, the access structure that is recognized by C. Therefore, to show how to construct a span program that computes Γ which is recognized by the circuit C, it is enough to show how to construct a span program that computes C.

Suppose we are given a circuit C that recognizes Γ and consists only of threshold gates, i. e. every gate k has l_k input wires and one output wire and the output is 1 if and only if at least t_k input wires carry a 1, where $1 \leq t_k \leq l_k$. Let R be a commutative ring with 1. We assume that R is l_k-good for every gate k in the circuit C. We have seen in theorem 3 that this can always be achieved. We know from proposition 4.6 that every gate k can be computed by an extended span program $(R, M_k, \mathbf{e}_1)$ with $M_k \in R^{l_k \times t_k}$. Our aim now is to combine these span programs to a span program $(R, M, \mathbf{e}_1)$ that computes the whole circuit C, and therefore, after changing the labelling, computes Γ. Let f be the output gate and $g_1, \ldots, g_n$ the subcircuits of C whose output wires are the input wires of f such that $C = f(g_1, \ldots, g_n)$. We will first show that from span programs computing f and $g_1, \ldots, g_n$, we can construct a span program computing C. ¿From this it will follow by induction that a span program computing C can be constructed from the $(R, M_k, \mathbf{e}_1)$.

In [8] Cramer, Damgård and Maurer showed, how a span program over a field computing C can be constructed given span programs computing f and g_i, respectively [3]. We are now going to show that their construction also works for extended span programs. So let $(R, F, \mathbf{e}_1)$ and $(R, G_i, \mathbf{e}_1)$, $i = 1, \ldots, n$, be extended span programs computing f and g_i, respectively. The following construction of an extended span program $(R, M, \mathbf{e}_1)$, that, as we will show later, computes C, is taken from [8].
Let x_{il} denote the l-th literal of g_i, for $i = 1, \ldots, n$, $l = 1, \ldots, m_i$. Let

[3] Their result also holds for span programs with multiplication.

$(\cdot, \ldots, \cdot, \ldots, \cdot)$ be a vector whose first location we number 0 and subsequent locations (i, k) with $1 \leq i \leq n$ and $1 \leq k \leq \text{rows}(F_i)$, where F_i consists of the rows of F that correspond to the i-th literal of f (i. e. indexed by i), hence there are $1 + \text{rows}(F)$ locations. The rows of M will be vectors whose locations are indexed as above. Next, let i, j, k be any integers satisfying $1 \leq i \leq n$, $1 \leq j \leq \text{rows}(G_i)$, $1 \leq k \leq \text{rows}(F_i)$.
The rows of M will consist of vectors $\mathbf{m}_{ijk}$ to be defined hereafter. Consider $(w_{ij}\mathbf{v}_{ik}, \mathbf{w}_{ij})$, where $\mathbf{v}_{ik}$ is the k-th row of F_i, and $(w_{ij}, \mathbf{w}_{ij})$ is the j-th row of G_i (i. e. w_{ij} is the first coordinate of that row and $\mathbf{w}_{ij}$ collects the remaining coordinates). The row $\mathbf{m}_{ijk}$ of M is constructed as follows. Place $w_{ij}\mathbf{v}_{ik}$ in the first location and $\mathbf{w}_{ij}$ in location (i, k). Fill up all other locations with $\mathbf{0}$. The dimensions of vectors placed in specific locations should be clear from the context. Note that the rows in M corresponding to the l-th literal x_{il} of function g_i are exactly those $\mathbf{m}_{ijk}$ where j is the index of a row in G_i associated with x_{il}.

Before we proof that $(R, M, \mathbf{e}_1)$ really computes h, we remark that using M to share an element s from a module E (with scheme 3) is equivalent to the following scheme. Use F to share s to n imaginary players and then for every share $\mathbf{s}_i$ use G_i to share all coordinates of $\mathbf{s}_i$ independently. From this it follows that if A is authorized, i. e. $h(A) = 1$, then $\mathbf{e}_1 \in M_A^t$. Indeed, if we use M to share a secret in, let say, R seen as a module over itself, then this is, as we have mentioned, equivalent to using first F and then G. Therefore if A is authorized, then the secret s can be calculated by a linear combination of the shares $\mathbf{s}_i = M_i\mathbf{x}$, $i \in A$. Thus there exists $\mathbf{v}_A$ such that $s = \langle \mathbf{v}_A, \mathbf{s}_A \rangle = \langle \mathbf{v}_A, M_A\mathbf{x} \rangle = \langle M_A^t \mathbf{v}_A, \mathbf{x} \rangle$. But also $s = \langle \mathbf{e}_1, \mathbf{x} \rangle$, hence $\langle M_A^t \mathbf{v}_A, \mathbf{x} \rangle = \langle \mathbf{e}_1, \mathbf{x} \rangle$. This holds for every $\mathbf{x}$, therefore $M_A^t \mathbf{v}_A = \mathbf{e}_1$.

Proposition 4.7 *The extended span program $(R, M, \mathbf{e}_1)$ constructed as above computes h. Further, the number of rows and columns of M are given by*

$$\begin{aligned} \text{rows}(M) &= \sum_{i=1}^{n} \text{rows}(F_i)\text{rows}(G_i) \\ \text{columns}(M) &= \text{columns}(F) + \sum_{i=1}^{n} \text{rows}(F_i)(\text{columns}(G_i) - 1) \end{aligned}$$

Proof. We have already seen that $h(A) = 1 \Rightarrow \mathbf{e}_1 \in \operatorname{Im} M_A^t$. Let now $A \subseteq \{(1,1), \ldots, (1, m_1), \ldots, (n, 1), \ldots, (n, m_n)\}$ such that $h(A) = 0$. We will show that there exists $\mathbf{a} \in \operatorname{Ker} M_A$, whose first coordinate is a unit. For $B = \{i \in \{1, \ldots, n\} \mid g_i(A) = 1\}$ we have $f(B) = 0$. Therefore, if F_B consists of the rows of F that are indexed by a $i \in B$, there exists $\mathbf{a}_B \in \operatorname{Ker} F_B$ such that the first coordinate of $\mathbf{a}_B$ is a unit, wlog say 1. Let now $i \notin B$, i. e. $g_i(A) = 0$. There exists $\mathbf{a}_i$ in the kernel of the matrix consisting of the rows of G_i that belong to A, such that the first coordinate is a unit, wlog say $\mathbf{a}_i = (1, \mathbf{b}_i)$. Let $\mathbf{a}$ be the vector having $\mathbf{a}_B$ in the first location, $\langle \mathbf{v}_{ik}, \mathbf{a}_B \rangle \mathbf{b}_i$ in the locations (i, k), $i \notin B$, $k = 1, \ldots, \operatorname{rows}(F_i)$, and $\mathbf{0}$ in the locations (i, k), $i \in B$, $k = 1, \ldots, \operatorname{rows}(F_i)$. Remember, $\mathbf{v}_{ik}$ is the k-th row of F_i. It is clear from construction that the first coordinate of $\mathbf{a}$ is a unit, namely 1. We will now show that $\mathbf{a} \in \operatorname{Ker} M_A$. Every coordinate of $M_A \mathbf{a}$ is of the form $\langle \mathbf{m}_{ijk}, \mathbf{a} \rangle$, where $\mathbf{m}_{ijk} = (w_{ij} \mathbf{v}_{ik}, 0, \ldots, 0, \mathbf{w}_{ij}, 0, \ldots)$ belongs to A and $\mathbf{w}_{ij}$ is in location (i, k). If $i \in B$, then

$$\langle \mathbf{m}_{ijk}, \mathbf{a} \rangle = w_{ij} \langle \mathbf{v}_{ik}, \mathbf{a}_B \rangle = 0$$

as $\mathbf{a}_B \in \operatorname{Ker} F_B$ and $\mathbf{v}_{ik}$ are the rows of F_B. If $i \notin B$, then

$$\begin{aligned} \langle \mathbf{m}_{ijk}, \mathbf{a} \rangle &= w_{ij} \langle \mathbf{v}_{ik}, \mathbf{a}_B \rangle + \langle \mathbf{w}_{ij}, \mathbf{b}_i \rangle \langle \mathbf{v}_{ik}, \mathbf{a}_B \rangle \\ &= (w_{ij} + \langle \mathbf{w}_{ij}, \mathbf{b}_i \rangle) \langle \mathbf{v}_{ik}, \mathbf{a}_B \rangle \\ &= \langle (w_{ij}, \mathbf{w}_{ij}), \mathbf{a}_i \rangle \langle \mathbf{v}_{ik}, \mathbf{a}_B \rangle \\ &= 0 \end{aligned}$$

as $\mathbf{a}_i$ is in the kernel of the matrix consisting of the rows of G_i belonging to A, which are exactly the $(w_{ij}, \mathbf{w}_{ij})$ we are here looking at. So it is proven that $(R, M, \mathbf{e}_1)$ computes h. The bounds on the number of rows and columns follow immediately from the construction of M.

□

When for the circuit C the expression $k \in C$ means that k is a gate of C and $\iota(C)$ stands for the number of input wires of C, then we can prove now

Theorem 6 *Let Γ be an access structure over $\mathcal{P} = \{1, \ldots, l\}$ recognized by a circuit C such that every gate $k \in C$ is an (t_k, l_k)-threshold gate for some $1 \leq t_k \leq l_k$, and let R be a commutative ring with 1. Then there exists an extended span program $(\bar{R}, M, \mathbf{e}_1)$ that computes Γ over some extension ring $\bar{R}$ of R. Furthermore, the number of rows and columns of M are bounded by*

$$\begin{aligned} \operatorname{rows}(M) &\leq \iota(C) \\ \operatorname{columns}(M) &\leq \iota(C) - \sum_{k \in C} (l_k - t_k) \end{aligned}$$

Thus, if $\iota(C)$ is bounded by a polynomial in l [4]*, the number of players, then $(\bar{R}, M, \mathbf{e}_1)$ is efficient.*

Proof. According to theorem 3 there exists an extension ring $\bar{R}$ of R which is l_k-good for every gate $k \in C$. We prove the theorem by induction on m, the number of gates in C. If $m = 0$, i. e. C consists only of a wire which is input and output wire at the same time, thus $\iota(C) = 1$, or, in other words, C is the circuit $C(x) = x$, then $M = 1$, seen as a 1×1 matrix, computes C and is bounded as claimed. If $m > 0$, then C can be written as $C = f(g_1, \ldots, g_n)$, where $f \in C$ is the final gate and $g_1, \ldots, g_n$ are the subcircuits of C whose output wires are the input wires of f. Gate f is a (t_f, l_f)-threshold gate with $l_f = n$. According to proposition 4.6 there exists an extended span program $(\bar{R}, F, \mathbf{e}_1)$ with $F \in \bar{R}^{l_f \times t_f}$ that computes f. For every $i \in \{1, \ldots, n\}$, the subcircuit g_i has less gates than C, therefore, by induction, there exists $(\bar{R}, G_i, \mathbf{e}_1)$ computing g_i such that $\mathrm{rows}(G_i) \leq \iota(g_i)$ and $\mathrm{columns}(G_i) \leq \iota(g_i) - \sum_{k \in g_i}(l_k - g_k)$. Proposition 4.7 implies that there exists a span program $(\bar{R}, M, \mathbf{e}_1)$ which computes $C = f(g_1, \ldots, g_n)$, and, after relabelling, the access structure Γ, and which is bounded by

$$\mathrm{rows}(M) \leq \sum_{i=1}^{n} \mathrm{rows}(F_i)\mathrm{rows}(G_i) = \sum_{i=1}^{n} \mathrm{rows}(G_i) \leq \sum_{i=1}^{n} \iota(g_i) = \iota(C)$$

and

$$\begin{aligned}
\mathrm{columns}(M) &\leq \mathrm{columns}(F) + \sum_{i=1}^{n} \mathrm{rows}(F_i)(\mathrm{columns}(G_i) - 1) \\
&= t_f + \sum_{i=1}^{n} \mathrm{columns}(G_i) - l_f \\
&= \sum_{i=1}^{n} \iota(g_i) - \sum_{i=1}^{n} \sum_{k \in g_i}(l_k - t_k) - (l_f - t_f) \\
&= \iota(C) - \sum_{k \in C}(l_k - t_k)
\end{aligned}$$

Note, F_i consists of only one row.

□

[4]This, of course, only makes sense if we are looking at a whole family of circuits, indexed by l.

Chapter 5

Security against Active Cheaters

Until now we always assumed that every participant of a secret sharing scheme does exactly what the scheme asks him to do. All the schemes presented so far fail if, for instance, some of the shareholders play faulty and give incorrect shares in the reconstruction phase, i. e. not the shares received from the dealer.

In the first section of this chapter we will modify scheme 3 in such a manner that it allows a set of players $A = B \cup C$, $B \cap C = \emptyset$, to reconstruct the shared secret, even if the players of C, we will call them *corrupted* players, give incorrect shares and the others, the players in B which we will call *honest* players, do not know which shares are correct and which are not. Of course this can only work if there are not too many corrupted and enough honest players. A secret sharing scheme for Γ with this property for B and C bounded by

$$\begin{aligned} &C \notin \Gamma \qquad \text{and} \\ &\forall D \subseteq B, D \notin \Gamma : B \setminus D \in \Gamma \end{aligned} \tag{5.1}$$

is called *robust*.
If Γ is a threshold access structure with threshold t, then these bounds mean that $|C| \leq t-1$ and $|B| \geq 2t-1$. If $B \cup C = \mathcal{P}$, which means that

all players are supposed to take part in the secret reconstruction (and the honest players do), then the second condition of (5.1) follows from the first if the access structure Γ has the property that $A, A', A'' \notin \Gamma \Rightarrow A \cup A' \cup A'' \neq \mathcal{P}$ (see also [12] and [8]).

In the second section of this chapter we will even allow the dealer to play faulty, and we will show how the players can by communication between each other detect that the dealer cheated or at least agree on a set of correct shares. And this even if some of the players play faulty as well. A *verifiable secret sharing scheme* is a robust scheme such that—even if there might be a faulty dealer and $\mathcal{P}$ contains, besides an authorized subset of players, an unauthorized subset of corrupted players—after the reconstruction phase the honest players have consistent shares defining some secret which is equal to the dealers secret if he is honest. The robustness guarantees that this secret can be reconstructed by a set $A = B \cup C$ as above, B and C bounded as in 5.1.

During the whole chapter, E again denotes an efficient finite module over a commutative ring R with 1 such that linear equation systems can be solved in R.
The following scheme 4 and the secret verification protocol in section 5.2 are taken from [8] and adapted to our somewhat more general situation. For the secret verification see also [1].

5.1 Robust Secret Sharing

Let Γ be an access structure computed by an extended span program $(R, M, \mathbf{e}_1)$, where $M \in R^{d \times e}$ and $\mathbf{e}_1 = (1, 0, \ldots, 0) \in R^e$. We will extend scheme 3 to a robust secret sharing scheme.
If the honest players could find out somehow which shares are correct and which are not, they could ignore the incorrect shares and just use the correct ones to reconstruct the secret. This is achieved by the following scheme.

Scheme 4

Distribution phase. $(R, M, \mathbf{e}_1)$ is public knowledge. Let $s \in E$ be the secret. The dealer chooses at random a symmetric matrix

$X \in E^{e \times e}$ with the only restriction that the upper-left corner of X contains s and sends the matrix $U_i = M_i X$ privately to player $i \in \mathcal{P}$. The actual share $\mathbf{s}_i$ of player i is the first column of U_i.

Reconstruction phase. Let $A = B \cup C$ be the set of players who want to reconstruct the secret such that B, the honest players, and C, the corrupted players, fulfil (5.1). Every player $i \in A$ broadcasts U_i. Let $\tilde{U}_i$ be what the player $i \in A$ claims to be U_i and $\tilde{\mathbf{s}}_i$ the first column of $\tilde{U}_i$. Every player computes for all $i \in A$ $D_i = \{j \in A \mid M_j \tilde{U}_i^t \neq (M_i \tilde{U}_j^t)^t\}$ and $D = \{i \in A \mid D_i \notin \Gamma\}$.
The secret s can now be reconstructed as in scheme 3 using the shares $\mathbf{s}_i$ with $i \in D$.

The following proposition ensures that all corrupted players who give an incorrect share $\mathbf{s}_i$ are detected in the above reconstruction phase.

Proposition 5.1 *The set D computed in the reconstruction phase of scheme 4 contains all the honest players but no corrupted player who gave a wrong share $\mathbf{s}_i$.*

Note that if player $i \in A$ is corrupted, i. e. $\tilde{U}_i \neq U_i$, but $\tilde{\mathbf{s}}_i = \mathbf{s}_i$, then it does not matter if he is detected or not.
For the proof of proposition 5.1 we need

Lemma 5.2 *For two vectors $\mathbf{x}, \mathbf{y} \in E^e$ with different first coordinates x_1 and y_1, $M_j\mathbf{x} = M_j\mathbf{y}$ holds at most for an unauthorized subset of players j.*

Proof. Let $D = \{j \in \mathcal{P} \mid M_j\mathbf{x} = M_j\mathbf{y}\}$. If $D \in \Gamma$, which means that $\mathbf{e}_1$ can be written as $\mathbf{e}_1 = M_D^t \mathbf{v}_D$ for a vector $\mathbf{v}_D$, then

$$
\begin{aligned}
x_1 - y_1 &= \langle \mathbf{e}_1, \mathbf{x} - \mathbf{y} \,] \\
&= \langle M_D^t \mathbf{v}_D, \mathbf{x} - \mathbf{y} \,] \\
&= \langle \mathbf{v}_D, M_D(\mathbf{x} - \mathbf{y}) \,] \\
&= 0
\end{aligned}
$$

which is a contradiction.

□

Proof of proposition 5.1. As $(M_i U_j^t)^t = U_j M_i^t = M_j X M_i^t = M_j U_i^t$, it follows straight from the restriction on C, i. e. from $C \notin \Gamma$, that if $i \in A$ is honest, then $D_i \notin \Gamma$.
Let now $i \in A$ be corrupted and $\tilde{\mathbf{s}}_i \neq \mathbf{s}_i$. Applying lemma 5.2 to the different columns of $\tilde{U}_i^t$ and U_i^t we can conclude that $M_j \tilde{U}_i^t = M_j U_i^t$ holds at most for an unauthorized subset of honest players j. ¿From this and from the lower boundary (5.1) of B it follows that $M_j \tilde{U}_i^t$ differs from $M_j U_i^t$ for an authorized subset of honest players j. But as $M_j U_i^t = (M_i U_j^t)^t = (M_i \tilde{U}_j^t)^t$ for honest players j, this means $D_i \in \Gamma$.

□

Proposition 5.3 *This modified scheme is still perfect, i. e. if $A \notin \Gamma$, then the U_i, $i \in A$, give away no information about the secret s.*

This claim seems pretty obvious as, compared to scheme 3, the additional information the players get, i. e. the U_i minus the first column $\mathbf{s}_i$, written as $U_i \setminus \mathbf{s}_i$, does not depend on the secret s. Still, we have to be careful, because both $\mathbf{s}_i$ and $U_i \setminus \mathbf{s}_i$ being independent of s does not necessarily imply that $U_i = (\mathbf{s}_i, U_i \setminus \mathbf{s}_i)$ is independent of s as well.

Proof. Let $A \notin \Gamma$, thus there exists $\mathbf{a} = (\lambda_1, \ldots, \lambda_e) \in \operatorname{Ker} M_A \subseteq R^e$ with $\lambda_1 = 1$. Let $\mathbf{a} \otimes \mathbf{a}$ denote the e by e matrix $(\lambda_i \lambda_j)$, i. e. the matrix whose i-th column is $\lambda_i \mathbf{a}$. Note, $\mathbf{a} \otimes \mathbf{a}$ is symmetric and $M_A(\mathbf{a} \otimes \mathbf{a}) = (0)$, the zero matrix. If U_A denotes the superposition of the U_i, i. e. $U_A = M_A X$ where X is the symmetric matrix with s in the upper-left corner, then the symmetric matrices X' fulfiling the equation $M_A X' = U_A$ are given by $X + Y$ where Y is symmetric and $M_A Y = (0)$. As for every $y \in E$ there exists such a Y with y in the upper-left corner, namely $\mathbf{a} \otimes \mathbf{a} \cdot y = (\lambda_i \lambda_j y)$, the equation $M_A X' = U_A$ has the same number of symmetric solutions for every possible choice of $s' \in E$ in the upper-left corner of X'.

□

This proves

Theorem 7 *If the access structure Γ is computed by an efficient extended span program over R, then there exists an efficient, robust and perfect secret sharing scheme over E for Γ.*

In the next section we will need the following generalization of proposition 5.1.

Proposition 5.4 *If in the distribution phase of scheme 4 the matrix X is not necessarily chosen symmetrical but in such a way that $M_jU_i^t = (M_iU_j^t)^t$ for all honest players $i, j \in \mathcal{P}$, then the set D computed in the reconstruction phase of scheme 4 still contains all the honest players and no corrupted player who gave a wrong share $\mathbf{s}_i$.*

Proof. As in the proof of proposition 5.1 it follows that $D_i \notin \Gamma$ if $i \in A$ is honest.
To show that $D_i \in \Gamma$ if $i \in A$ is corrupted and $\tilde{\mathbf{s}}_i \neq \mathbf{s}_i$, we first claim that for an honest player $i \in \mathcal{P}$ the first column of M_iX coincides with the first column of M_iX^t. Indeed, as for a fix honest player i

$$M_j(M_iX)^t = M_jU_i^t = (M_iU_j^t)^t = M_jXM_i^t = M_j(M_iX^t)^t$$

holds for all honest players $j \in \mathcal{P}$ and the honest players are an authorized set, the claim follows straight from lemma 5.2.
Let now $i \in A$ be corrupted and $\tilde{\mathbf{s}}_i \neq \mathbf{s}_i$. Consider the subset of honest players $B_i = \{j \in B \mid M_j\tilde{U}_i^t = (M_iU_j^t)^t\}$. If B_i is authorized, i. e. $\mathbf{e}_1$ can be written as $\mathbf{e}_1 = M_{B_i}^t\mathbf{v}_{B_i}$, then $\tilde{\mathbf{s}}_i = \tilde{U}_i\mathbf{e}_1 = \tilde{U}_iM_{B_i}^t\mathbf{v}_{B_i} = (M_{B_i}\tilde{U}_i^t)^t\mathbf{v}_{B_i} = M_iU_{B_i}^t\mathbf{v}_{B_i} = M_iX^tM_{B_i}^t\mathbf{v}_{B_i} = M_iX^t\mathbf{e}_1 = M_iX\mathbf{e}_1 = \mathbf{s}_i$ which is a contradiction and therefore $B_i \notin \Gamma$. From the lower boundary 5.1 for B it thus follows that $M_j\tilde{U}_i^t \neq (M_iU_j^t)^t$ for an authorized subset of (honest) players j and therefore $D_i \notin \Gamma$.

□

5.2 Verifiable Secret Sharing

Now we will extend scheme 3 even further such that if there is a faulty dealer, then the honest players can correct the inconsistent shares. For this let again Γ be an access structure computed by an extended span program $(R, M, \mathbf{e}_1)$, and let E be a module over R. Further, we assume that some dealer—honest or faulty—has distributed matrices U_i according to scheme 4, supposedly of the form $U_i = M_iX$ for a fix symmetric matrix X.
Note that two players i and j can in a way compare their shares by checking if $M_jU_i^t = (M_iU_j^t)^t$. If $M_jU_i^t \neq (M_iU_j^t)^t$, then truly something is wrong. Either one of the players is corrupted, i. e. U_i or U_j is not what player i or

j, respectively, got from the dealer, or U_i and U_j are not consistent, which means that the dealer is faulty.
The following *secret verification protocol*, carried out between the players after the distribution phase, provides that all honest players get consistent shares of a secret if, besides the dealer, there is only an unauthorized set of corrupted but an authorized set of honest players.

> *Step 1.* Every two players i and j check if $M_j U_i^t = (M_i U_j^t)^t$ by privately exchanging these values. If a player i did not receive U_i from the dealer (in the right format), he uses a default value.
>
> *Step 2.* For each player i, if player i finds disagreements in values received from an authorized set of players, then the dealer is clearly faulty and player i broadcasts an *accusation* against the dealer, asking him to broadcast U_i. If there are disagreements only in values received from an unauthorized set of players, player i broadcasts a request to the dealer to make public those $M_j U_i$, $j \in \mathcal{P}$, that did not agree with the received values.
>
> *Step 3.* The dealer has to response to all the accusations and complaints by making public all the values he has been asked for. If then some player i observes that some new public information contradicts the U_i he is holding, he accuses the dealer asking him to broadcast U_i. Again the dealer has to response to all accusations, and this goes on like this until no accusation is made anymore.
>
> *Step 4.* If at this point an authorized set of players have accused the dealer, the dealer is clearly faulty and all the players take a fixed default set of shares to represent the dealers secret. Likewise, if the dealer did not answer all the broadcasted requests or if the public information contradicts itself he is declared faulty. Otherwise, the complaining players take the public information as their shares.

It is easy to see that if the dealer shares his secret correctly, then the corrupted players do not learn anything new from this communication and the shares held by honest players consistently determine s, the dealers secret. This proves one part of

Proposition 5.5 *As long as the set of honest players is authorized and the set of corrupted players is unauthorized, the above protocol results in the correct players holding consistent shares* $\mathbf{s}_i$ *of some secret* $s \in E$.

Proof. Because of the above remark, we have only to look at the case where the dealer plays faulty. If the dealer is recognized to play faulty, so that the players take the default set of shares, the theorem is clearly true. Otherwise, the honest players, say B, end up with matrices U_i, either received in the distribution phase or after accusing the dealer, such that $M_j U_i^t = (M_i U_j^t)^t$ for all i and j in B. Indeed, if there still was some disagreement between honest players, the protocol would not be finished. As $B \in \Gamma$, there exists $\mathbf{v}_B$ with $M_B^t \mathbf{v}_B = \mathbf{e}_1$. Therefore, for $i \in B$

$$\mathbf{s}_i = U_i \mathbf{e}_1 = U_i M_B^t \mathbf{v}_B = (M_B U_i^t)^t \mathbf{v}_B = M_i U_B^t \mathbf{v}_B$$

Thus, the $\mathbf{s}_i$ form a consistent set of shares for the players in B.

□

As the above secret verification only provides consistent shares $\mathbf{s}_i$ but not necessarily consistent matrices U_i, i. e. matrices of the form $U_i = M_i X$ for a fixed matrix X chosen as in the distribution phase of scheme 4, we can only guarantee that the secret, determined by the $\mathbf{s}_i$, can efficiently be computed if there are no corrupted players in the reconstruction phase. But as we assumed that there are cheaters in the secret verification, this is not a realistic scenario.

In the following scheme, the secret can be reconstructed efficiently even if there are corrupted players in the reconstruction phase.

Scheme 5

Distribution phase. $(R, M, \mathbf{e}_1)$ is public knowledge. Let $s \in E$ be the secret. The dealer chooses the matrix X as in scheme 4. Further, he chooses random symmetric matrices $X_j \in R^{e \times e}$, $j = 1, \ldots, e$, with the restriction that the first column of X_j is equal to the j-th column of X. He then sends the matrices $U_i = M_i X$ and $U_{ij} = M_i X_j$, $j = 1, \ldots, e$, to player i.

> To verify the secret, every player $i \in \mathcal{P}$ checks if the first column of U_{ij} is equal to the j-th column of U_i for $j = 1, \ldots, e$. If a player finds disagreement, he accuses the dealer. Further, the players apply the above secret verification protocol to the matrices $U_i, U_{i1}, \ldots, U_{ie}$. If a player $i \in \mathcal{P}$ accuses the dealer, the dealer has to make public i's matrices $U_i, U_{i1}, \ldots, U_{ie}$.
>
> *Reconstruction phase.* The secret can now be reconstructed as in scheme 4 using the matrices U_i.

It is clear from proposition 5.5 that the $\mathbf{s}_i$, the first columns of the U_i, define a secret s. Further, if the dealer is honest, then s is the secret he shared. The following proposition, together with proposition 5.4, guarantees that the secret can be recovered even if some players play faulty.

Proposition 5.6 *The U_i of the honest players, either received in the beginning of the distribution phase or after having accused the dealer, are of the form $U_i = M_i X$ for a fixed matrix X and fulfil $M_j U_i^t = (M_i U_j^t)^t$.*

Proof. Proposition 5.5 guarantees that the first column of U_{ij}, and therefore the j-th column of U_i, is of the form $M_i \mathbf{x}_j$. Hence, $U_i = M_i X$ for the matrix $X = (\mathbf{x}_1, \ldots, \mathbf{x}_e)$. The equality $M_j U_i^t = (M_i U_j^t)^t$ for honest players i and j follows from the fact that there are no accusations against the dealer anymore. □

We have to ensure that this scheme is still perfect.

Proposition 5.7 *If A is an unauthorized subset of players, then the matrices $U_i, U_{i1}, \ldots, U_{ie}$ for $i \in A$ reveal nothing about the secret.*

Proof. Let A be an unauthorized subset. Consider the linear function that maps an $(e+1)$-tuple of matrices $(X, X_1, \ldots, X_e)$ as in the scheme, i. e. all symmetrical and the first column of X_j equals to the j-th column of X, to the $(e+1)$-tuple of matrices $(M_A X, M_A X_1, \ldots, M_A X_e)$. To prove the proposition it is sufficient to show that for every element $y \in E$ there exists a $(e+1)$-tuple $(Y, Y_1, \ldots, Y_e)$ in the kernel of this linear function such that y is in the upper-left corner of Y. Let $\mathbf{a} = (\lambda_1, \ldots, \lambda_e) \in \operatorname{Ker} M_A \subseteq R^e$ with $\lambda_1 = 1$. $\mathbf{a} \otimes \mathbf{a}$ is defined as in the proof of proposition 5.3 as the matrix $(\lambda_i\lambda_j)$ and $\mathbf{a}\otimes\mathbf{a}\cdot y$ as the matrix $(\lambda_i\lambda_j y)$. Now it is easy to see that $Y=\mathbf{a}\otimes\mathbf{a}\cdot y, Y_1=\lambda_1\mathbf{a}\otimes\mathbf{a}\cdot y,\ldots,Y_e=\lambda_e\mathbf{a}\otimes\mathbf{a}\cdot y$ is exactly what we are looking for. Remember that $\lambda_1=1$.

□

Therefore, we have proven

Theorem 8 *If the access structure* Γ *is computed by an efficient extended span program over* R*, then there exists an efficient, verifiable and perfect secret sharing scheme over* E *for* Γ*.*

Chapter 6

Application: RSA Function Sharing

If the secret key of a digital signature scheme, for instance RSA [20], is shared among a set of people with some secret sharing scheme, then an authorized subset of these people can reconstruct the secret key and sign any message, whereas an unauthorized subset cannot. But after one such signature has been realized, the secret key is known to every single person of this authorized subset and even to others who have silently listened to the key reconstruction. So everyone of these persons can now sign messages on their own. In this chapter we will present an RSA function sharing scheme which enables an authorized subset of people to sign messages without computing the secret key. And this even if some players play faulty and try to sabotage the signing.

In [9] de Santis *et al.* investigated threshold function sharing and gave sufficient conditions for functions to be sharable. Especially, they presented a threshold scheme for the RSA function [20]. Using their scheme and employing extensions developed of [18, 19] and [5], Gennaro *et al.* developed in [11] a robust (and even verifiable) (t, l)-threshold RSA function sharing scheme.

We will now use our results about secret sharing schemes based on span programs to build up an RSA function sharing scheme for any access structure which is computed by a span program. Further, we will extend this to a robust scheme. We have to be aware that the robustness we achieve is somewhat weaker than the robustness achieved by Gennaro *et al.* in [11]. If we apply our scheme to a (t, l)-threshold access structure, we have to demand that, besides not more than $t-1$ corrupted, at least $2t-1$ honest players are present in the reconstruction phase, whereas in [11] only t honest players are needed.

6.1 Definitions

Let Γ be an access structure over the set of players $\mathcal{P} = \{1 \ldots, l\}$. Further, let F be some set of functions with common domain. A *function sharing scheme* for Γ contains two algorithms, the first, the *distribution* algorithm, which will be executed by the dealer, takes as input a function f in F and outputs, besides some public information, l *share functions* $f_1, \ldots, f_l$, each being privately sent to the corresponding player, such that any authorized subset of players can compute the function evaluation $f(x)$ for any x in the domain of f from their *partial function evaluations* $f_i(x)$ using the second algorithm, the *evaluation* algorithm, yet an unauthorized subset of players cannot, using any method.

We call a function sharing scheme *perfect* if for any unauthorized subset A the following property is fulfiled. Given a set of tuples $(x_j, f(x_j))$, then the corresponding partial function evaluations $f_i(x_j)$ of all players $i \in \mathcal{P}$, the share functions f_i of the players $i \in A$ plus the public information give away no more information about $f(x)$ for a fix x as the $(x_j, f(x_j))$ alone.

The definition of an *efficient* function sharing scheme is analogue to the one of an efficient secret sharing scheme given in chapter 2. That means, we are looking at a whole family of access structures, indexed by the number of players l, and a whole family of sets of functions, indexed by κ, and the distribution and the evaluation algorithm must run in polynomial time in $\max(l, \log \kappa)$.

Whereas a secret sharing scheme is a one-time operation in the sense that after the reconstruction phase the secret is revealed, in a function sharing scheme the secret function is never revealed and therefore reusable many times.

Analogue to secret sharing schemes we can define zero-knowledge and minimal-knowledge function sharing schemes. A function sharing scheme is called

(statistical) zero-knowledge if the following holds. Any unauthorized subset A can, with their knowledge before the function distribution, construct a in $\max(l, \log \kappa)$ polynomial time algorithm that takes a polynomial number of tuples $(x_j, f(x_j))$ as input and outputs simulated share functions $\tilde{f}_i$ for the players $i \in A$, partial function evaluations $\tilde{f}_i(x_j)$ for all $i \in \mathcal{P}$ and x_j plus public information, everything (statistical) indistinguishable from the corresponding information generated in the scheme for the function f. It is clear that a zero-knowledge function sharing scheme must be perfect.
A function sharing scheme is called *(statistical) minimal-knowledge* if any authorized subset A can, with their knowledge before the distribution, construct a polynomial time algorithm which takes a function f and a polynomial number of elements x_j as input and outputs simulated share functions $\tilde{f}_i$ for the players $i \in A$, partial function evaluations $\tilde{f}_i(x_j)$ for all players $i \in \mathcal{P}$ and public information, all (statistical) indistinguishable from those generated in the scheme for the same function f.
A zero-knowledge and minimal-knowledge function sharing scheme guarantees that the players of a subset—authorized or unauthorized—get only that amount of information they are supposed to get and nothing more.

The basic idea for sharing an RSA function $m \mapsto m^s$ is the following. We share the secret exponent s using a secret sharing scheme. In the evaluation phase, every participant i computes from his share s_i and the function input m the partial function evaluation $c_i = m^{s_i}$ such that $c = m^s$ can be computed from the c_i.
Of course, such an RSA function sharing scheme can only be as secure as RSA itself. Indeed, if an adversary is able to compute the exponent s from the values m and $c = m^s$, then the function is revealed after one evaluation, even though the scheme is be perfect. But if we assume that RSA cannot be broken efficiently in the sense that there exists no polynomial time algorithm which computes m^s from a polynomial numbers of tuples $(m_1, m_1^s), (m_2, m_2^s), \ldots$, all m_i different from m, then a zero-knowledge and minimal-knowledge function sharing scheme guarantees that no unauthorized subset can compute (in polynomial time) $c = m^s$ for a new m, even after having listened to (a polynomial number of) function evaluations being computed with the evaluation algorithm.

6.2 Preliminary

During the whole chapter let n be the RSA composite, i. e. $n = pq$ for two large primes p and q, and $\varphi(n) = (p-1)(q-1)$, the order of $\mathbb{Z}_n^*$. The group $\mathbb{Z}_{\varphi(n)}$ can be seen as a $\mathbb{Z}$-module. If for a given access structure Γ there exists an (efficient) extended span program $(\mathbb{Z}, M, \mathbf{e}_1)$ that computes Γ, then the secret exponent $s \in \mathbb{Z}_{\varphi(n)}$ can be shared using M. But it might be that, to get an appropriate span program, we have to extend the ring $\mathbb{Z}$ and therefore the module $\mathbb{Z}_{\varphi(n)}$, as in sections 3.2 and 3.3, to $\mathbb{Z}[u]$ and $(\mathbb{Z}_{\varphi(n)})^m$, respectively, where u is a zero of a irreducible monic polynomial $f(X) = f_0 + \cdots + f_{m-1}X^{m-1} + X^m \in \mathbb{Z}[X]$, $f_m \neq 0$, for instance the cyclotomic polynomial $f(X) = 1 + X + \cdots + X^{r-1}$ for a prime r. We will denote $R = \mathbb{Z}[u]$ and $E = (\mathbb{Z}_{\varphi(n)})^m$ and, further, $H = \mathbb{Z}_n^*$. For $h \in H$ and $\mathbf{x} = (x_1, \ldots, x_m) \in E$ we define the $\mathbf{x}$-th power of h as

$$h^{\mathbf{x}} = (h^{x_1}, \ldots, h^{x_m}) \in H^m$$

Note that as $\varphi(n)$ is the order of H, h^{x_i} is well defined for $x_i \in \mathbb{Z}_{\varphi(n)}$. We have to be aware that $h^{\mathbf{x}}$ is not an element of H but of H^m. If multiplication in H^m is meant component-wise we can further define for $\mathbf{h} = (h_1, \ldots, h_m) \in H^m$ and $\lambda = \lambda_0 + \lambda_1 u + \cdots + \lambda_{m-1}u^{m-1} \in R$ the λ-th power of $\mathbf{h}$, $\mathbf{h}^\lambda \in H^m$, as follows.

$$\mathbf{h}^\lambda = \mathbf{h}^{\lambda_0}\mathbf{h}^{\lambda_1 u} \cdots \mathbf{h}^{\lambda_{m-1}u^{m-1}}$$

where recursively

$$\mathbf{h}^{\lambda_i u^i} = (\mathbf{h}^{\lambda_i u^{i-1}})^u$$

and

$$\begin{aligned} \mathbf{h}^{\lambda_i} &= (h_1^{\lambda_i}, \ldots, h_m^{\lambda_i}) \qquad \text{and} \\ \mathbf{h}^u &= (h_m^{-f_0}, h_1 h_m^{-f_1}, \ldots, h_{m-1}h_m^{-f_{m-1}}) \end{aligned}$$

See the similarity to the scalar multiplication on E, defined in section 3.3. We have to be aware that the elements of E act as powers on H whereas the elements of R act as powers on H^m. We leave it to the reader to prove that the following exponentiation rules hold.

$$\begin{array}{rclrcl} h^{\mathbf{x}+\mathbf{y}} & = & h^{\mathbf{x}}h^{\mathbf{y}} & \mathbf{h}^{\lambda+\mu} & = & \mathbf{h}^{\lambda}\mathbf{h}^{\mu} \\ (gh)^{\mathbf{x}} & = & g^{\mathbf{x}}h^{\mathbf{x}} & (\mathbf{gh})^{\lambda} & = & \mathbf{g}^{\lambda}\mathbf{h}^{\lambda} \\ h^{\lambda\mathbf{x}} & = & (h^{\mathbf{x}})^{\lambda} & \mathbf{h}^{\lambda\mu} & = & (\mathbf{h}^{\lambda})^{\mu} \\ h^{(0,\ldots,0)} & = & \mathbf{1} & \mathbf{h}^{0} & = & \mathbf{1} \\ h^{(1,\ldots,1)} & = & (h,\ldots,h) & \mathbf{h}^{1} & = & \mathbf{h} \end{array} \tag{6.1}$$

for all $g, h \in H$, $\mathbf{g}, \mathbf{h} \in H^m$, $\mathbf{x}, \mathbf{y} \in E$ and $\lambda, \mu \in R$.

As a final remark of this section we want to stress that the value m^s, $m \in \mathbb{Z}_n^*$ and $s \in \mathbb{Z}_{\varphi(n)}$, is the first entry of the tuple $m^{(s,0,\ldots,0)}$ with $(s, 0, \ldots, 0) \in E$. Hence, if we can find a scheme that shares the function $H \to H^m, m \mapsto m^{\mathbf{s}}$ for $\mathbf{s} \in E$, then we automatically have a function sharing scheme for the RSA function $H \to H, m \mapsto m^s$ for $s \in \mathbb{Z}_{\varphi(n)}$.

6.3 RSA Function Sharing without Cheaters

Let $R = \mathbb{Z}[u]$, $E = (\mathbb{Z}_{\varphi(n)})^m$ and $H = \mathbb{Z}_n^*$ as in the previous section. The coordinates of $E = (\mathbb{Z}_{\varphi(n)})^m$ will be represented by integers in $\{0, 1, \ldots, \varphi(n)-1\}$. The case $r = 1$ which yields to the special case $R = \mathbb{Z}$ and $E = \mathbb{Z}_{\varphi(n)}$ is allowed. In this section we will not worry anymore about the fact that the elements of R and E are polynomials in u and m-tuples but see them as formal objects of the ring R and the R-module E and use the exponentiation rules (6.1) when acting on H^m and H, respectively. Also, H^m with the component-wise multiplication can be seen as a formal group. Finally, let $(R, M, \mathbf{e}_1)$ be an extended span program computing an access structure Γ.

Before we can present the function sharing scheme we have to introduce some notation. For an element $h \in H$ and a matrix $U = (u_{jk})$ with entries in E we write

$$h^U = (h^{u_{jk}})$$

Note that the entries of the matrix h^U are elements from the group H^m. Further, if $K = (\kappa_{ij})$ and $S = (s_{jk})$ are matrices with entries in R and H^m, respectively, such that the number of columns of K is equal to the number of rows of S, then ${}^{K}S$ denotes the matrix

$$ {}^{K}S = (\prod_j s_{jk}^{\kappa_{ij}})_{ik} $$

having the same number of rows as K and the same number of columns as S. If we would write the exponentiation and the multiplication in H^m as (scalar) multiplication and addition, respectively, then ${}^{K}S$ would be the normal matrix multiplication KS.
The following rules are easy to verify. For all matrices S and T with entries in H^m, h^U as above and K and L with entries in R, all with suitable dimensions, it is

$$ \begin{aligned} {}^{K}(h^U) &= h^{KU} \\ {}^{\mathbf{e}_1{}^t}S &= (s_{11}, s_{12}, \ldots), \text{ the first row of } S \\ {}^{KL}S &= {}^{K}({}^{L}S) \\ {}^{K}(S \cdot T) &= {}^{K}S \cdot {}^{K}T \end{aligned} $$

where in the second equation $\mathbf{e}_1{}^t$ is a row vector and in the last equation the multiplication is meant component-wise.

Now we are equipped to present the scheme that shares the function $H \to H^m, m \mapsto m^s$ for every secret exponent $s \in E$.

Scheme 6

Distribution phase. Let $s \in E$ be the secret exponent. The dealer shares s as in scheme 3. For $i \in \mathcal{P}$ let $\mathbf{s}_i$ denote player i's share.

Evaluation phase. Let A be in Γ and $m \in H$ the input of the secret function. Further, let $\mathbf{v}_A$ be such that $M_A^t \mathbf{v}_A = \mathbf{e}_1$. Every player $i \in A$ computes and broadcasts the partial function evaluation $\mathbf{c}_i = m^{\mathbf{s}_i}$ and then computes $c = {}^{\mathbf{v}_A{}^t}\mathbf{c}_A$, where $\mathbf{c}_A$ is the superposition of the $\mathbf{c}_i$, $i \in A$.

Note that $R = \mathbb{Z}[u] \cong \mathbb{Z}[X]/<f(X)>$, therefore, according to corollary A.5, the vector $\mathbf{v}_A$ can be computed.

Proposition 6.1 *The value c computed in the evaluation phase of scheme 6 is equal to the function evaluation m^s.*

Proof. If $A \in \Gamma$ and $\mathbf{s}_A$ denotes the superposition of the $\mathbf{s}_i$, $i \in A$, then

$$c = {}^{\mathbf{v}_A{}^t}\mathbf{c}_A = {}^{\mathbf{v}_A{}^t}(m^{\mathbf{s}_A}) = m^{\mathbf{v}_A{}^t\mathbf{s}_A} = m^{\langle \mathbf{v}_A, \mathbf{s}_A]} = m^s$$

which is what we claimed.

□

Because a tuple message and signature, (m, m^s), already uniquely defines the secret s, the share functions and the partial function evaluations give no further information about s, hence, according to the definition given in section 6.1, scheme 6 is perfect. But this is of course not satisfactory. What we want is that the share functions of an unauthorized subset and the partial function evaluations cannot be used to *compute* something (efficiently) which cannot be computed from (m, m^s). So we want the scheme to be zero-knowledge. A necessary condition for zero-knowledge is that the shares (the share functions) of an unauthorized subset can be simulated. In the threshold case, this can be done by choosing random elements as mentioned in section 3.4, but it is easy to see that this simulator fails in the general case. Indeed, if, for instance, one row of M_i only consists of even numbers (i. e. of multiples of two), then the corresponding coordinate of the share must be even as well (as also $\varphi(n)$ is even). Further, if one row of M_i only consists of multiples of a positive integer k, but k does not divide the corresponding coordinate of the share, then $\varphi(n)$ cannot be a multiple of k. So it seems that scheme 6 is in fact not zero-knowledge, even though it seems to be secure in the sense that there exists no method which allows an adversary to compute $\varphi(n)$ efficiently. In the following section we present a modification of scheme 6 which is statistical zero-knowledge if the number of players is constant.

6.4 Zero-Knowledge RSA Function Sharing

Let $E' = \mathbb{Z}^m$, seen as a module over $R = \mathbb{Z}[u]$ with the scalar multiplication as in section 3.3. For $a \leq b \in \mathbb{Z}$ we define the *interval*

$$[a, b] = \{(x_1, \ldots, x_m) \in E' \mid a \leq x_i \leq b \text{ for } i = 1, .., m\} \subset E'$$

Note that the interval $[a, b]$ consists of $(b - a + 1)^m$ different elements.
Before we can present the scheme we have to introduce some kind of a *norm* in R and E'.

For $\lambda = \lambda_0 + \cdots + \lambda_{m-1}u^{m-1} \in R = \mathbb{Z}[u]$ and $x = (x_1, \ldots, x_m) \in E' = \mathbb{Z}^m$ we define $\|\lambda\|$ and $\|x\|$ as

$$\begin{aligned} \|\lambda\| &= \max_i |\lambda_i| \\ \|x\| &= \max_i |x_i| \end{aligned}$$

where $|\cdot|$ stands for the absolute value of an integer. It is clear that for an integer $a \geq 0$ the set $\{x \in E' \mid \|x\| \leq a\}$ coinsides with the interval $[-a, +a]$. Further, we have the following rules.

Lemma 6.2 *All $\lambda \in R$ and $x, y \in E'$ fulfil*

$$\begin{aligned} \|x + y\| &\leq \|x\| + \|y\| \\ \|\lambda x\| &\leq \|\lambda\| \|x\| \Delta \end{aligned}$$

where Δ only depends on $f(X)$.

Remember that $f(X) = f_0 + \cdots + f_{m-1}X^{m-1} + X^m$ is the irreducible polynomial of which u is a zero.

Proof. The first inequality should be clear. To prove the second, let $\lambda = \lambda_0 + \cdots + \lambda_{m-1}u^{m-1}$ and $\delta = \max |f_i|$. From the definition of the scalar multiplication on E' it follows that $\|\lambda_i x\| \leq |\lambda_i| \|x\|$ and $\|ux\| \leq (1 + \delta)\|x\|$ and therefore (for $x \neq 0$)

$$\begin{aligned} \|\lambda x\| &= \|(\lambda_0 + \cdots + \lambda_{m-1}u^{m-1})x\| \\ &\leq |\lambda_0| + |\lambda_1| \|ux\| + \cdots + |\lambda_{m-1}| \|u^{m-1}x\| \\ &\leq \|\lambda\| + \|\lambda\|(1 + \delta)\|x\| + \cdots + \|\lambda\|(1 + \delta)^{m-1}\|x\| \\ &= \|\lambda\| \left(1 + \|x\|(1 + \delta)\frac{1 - (1 + \delta)^{m-2}}{-\delta}\right) \\ &\leq \|\lambda\| \|x\| \left(1 - \frac{1 + \delta - (1 + \delta)^{m-2}}{\delta}\right) \\ &= \|\lambda\| \|x\| \frac{(1 + \delta)^{m-2} - 1}{\delta} \end{aligned}$$

which proves the second inequality if we put $\Delta = ((1 + \delta)^{m-2} - 1)/\delta$.

□

¿From now on, we again forget that the elements of R are polynomials in u and the elements of E' m-tuples and just use the fact that E' is a module over R, that the above proposition holds and that an interval $[a,b]$ contains $(b-a+1)^m$ different elements of E'.
For vectors $\mathbf{a} = (\lambda_1, \ldots, \lambda_e) \in R^e$ and $\mathbf{x} = (x_1, \ldots, x_e) \in E'^e$ we define

$$\begin{aligned} \|\mathbf{a}\| &= \max_i \|\lambda_i\| \\ \|\mathbf{x}\| &= \max_i \|x_i\| \end{aligned}$$

It is clear that for an integer $a \geq 0$ the set $\{\mathbf{x} \in E'^e \mid \|\mathbf{x}\| \leq a\}$ coinsides with the set $[-a, +a]^e$ and that the following corollary follows straight from the above proposition.

Corollary 6.3 *All $\lambda \in R$, $\mathbf{a} = (\lambda_1, \ldots, \lambda_e) \in R^e$, $x \in E'$ and $\mathbf{x}, \mathbf{y} \in E'^e$ fulfil*

$$\begin{aligned} \|\mathbf{x} + \mathbf{y}\| &\leq \|\mathbf{x}\| + \|\mathbf{y}\| \\ \|\lambda \mathbf{x}\| &\leq \|\lambda\| \|\mathbf{x}\| \Delta \\ \|\mathbf{a} \cdot x\| &\leq \|\mathbf{a}\| \|x\| \Delta \end{aligned}$$

where $\mathbf{a} \cdot x$ is defined as $\mathbf{a} \cdot x = (\lambda_1 x, \ldots, \lambda_e x)$.

It is clear that the following scheme is correct in the sense that an authorized subset can, from their shares, compute $c = m^s$ for any given m.

Scheme 7

Distribution phase. Let $s \in [0, \varphi(n) - 1]$ be the secret (or, to be precise, a representation of the secret). The dealer puts $\mathbf{x} = (s, x_2, \ldots, x_e)$ where the coordinates x_i are chosen at random from $[-n^2, +n^2] \subset E'$. Then, he privately distributes the shares $\mathbf{s}_i = M_i \mathbf{x} \in E'^{d_i}$.

Evaluation phase. Let A be in Γ and $m \in H$ the input of the secret function. The players in A compute the function evaluation $c = m^s$ as in scheme 6.

Proposition 6.4 *If the number of players l (and the access structure) is constant, then, for an unauthorized subset A, the simulation $\tilde{\mathbf{s}}_A = M_A \tilde{\mathbf{x}}$ with*

$\tilde{\mathbf{x}} = (0, x_2, \ldots, x_e)$ and the $x_i \in [-n^2, +n^2]$ chosen at random is statistical indistinguishable from the shares generated by the scheme for any secret s.

Proof. The claim is certainly true for the secret $s = 0$. For an arbitrary s, we know that $\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{a} \cdot s$ has s as first entry and solves $M_A \mathbf{x} = \tilde{\mathbf{s}}_A$, where $\mathbf{a} \in \operatorname{Ker} M_A$ with first coordinate 1. The probability that this $\mathbf{x}$ lies in $[-n^2, +n^2]^e$ is

$$\begin{aligned}
P[\,\|\mathbf{x}\| \leq n^2\,] &= P[\,\|\tilde{\mathbf{x}} + \mathbf{a} \cdot s\| \leq n^2\,] \\
&\geq P[\,\|\tilde{\mathbf{x}}\| \leq n^2 - \|\mathbf{a} \cdot s\|\,] \\
&\geq P[\,\|\tilde{\mathbf{x}}\| \leq n^2 - \|\mathbf{a}\| |s| \Delta\,] \\
&\geq P[\,\|\tilde{\mathbf{x}}\| \leq n^2 - \|\mathbf{a}\|(\varphi(n) - 1)\Delta\,] \\
&\geq P[\,\|\tilde{\mathbf{x}}\| \leq n^2 - n\|\mathbf{a}\|\Delta\,] \\
&= \left(\frac{2(n^2 - n\|\mathbf{a}\|\Delta) + 1}{2n^2 + 1}\right)^{m(e-1)} \\
&\geq \left(\frac{1}{1 + \frac{1}{2n^2}} - \frac{\|\mathbf{a}\|\Delta}{n + \frac{1}{2n}} + \frac{1}{2n^2 + 1}\right)^{m(e-1)}
\end{aligned}$$

As the numbers $\|\mathbf{a}\|$, Δ, m and e only depend on the number of players and the span program, this final value differs only by an exponentially (in $\log n$) small number from 1, which proves the claim.

□

Proposition 6.5 *If the number of players l (and the access structure) is constant, then scheme 7 is statistical zero-knowledge and perfect minimal-knowledge.*

Proof. It is clear that scheme 7 is minimal-knowledge, as an authorized subset can use the scheme itself as a simulator. To show that it is statistical zero-knowledge, let A be an unauthorized subset. The players in A can simulate their shares $\tilde{\mathbf{s}}_A = M_A \tilde{\mathbf{x}}$ according to the previous proposition. Let $m \in H$ and $c = m^s$. We have to show that, knowing (m, c), the players in A can simulate the corresponding partial function evaluations c_i. For this let $\mathbf{a} = (\lambda_1, \ldots, \lambda_e) \in \operatorname{Ker} M_A$ with the first coordinate 1. We know that

$\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{a} \cdot s$ fulfils $M_A \mathbf{x} = \tilde{\mathbf{s}}_A$, has s in the first location and is not in $[-n^2, +n^2]$ with only an exponentially small probability. Therefore,

$$\begin{aligned}
\tilde{\mathbf{c}}_{\mathcal{P}\setminus A} &= m^{M_{\mathcal{P}\setminus A}\mathbf{x}} \\
&= m^{M_{\mathcal{P}\setminus A}\tilde{\mathbf{x}}}\; m^{M_{\mathcal{P}\setminus A}(\mathbf{a}\cdot s)} \\
&= m^{M_{\mathcal{P}\setminus A}\tilde{\mathbf{x}}}\; {}^{M_{\mathcal{P}\setminus A}}\!\left(m^{\mathbf{a}\cdot s}\right) \\
&= m^{M_{\mathcal{P}\setminus A}\tilde{\mathbf{x}}}\; {}^{M_{\mathcal{P}\setminus A}}\!\left(m^{(\lambda_1 s,\ldots,\lambda_e s)}\right) \\
&= m^{M_{\mathcal{P}\setminus A}\tilde{\mathbf{x}}}\; {}^{M_{\mathcal{P}\setminus A}}\!\left(m^{\lambda_1 s},\ldots,m^{\lambda_e s}\right) \\
&= m^{M_{\mathcal{P}\setminus A}\tilde{\mathbf{x}}}\; {}^{M_{\mathcal{P}\setminus A}}\!\left(c^{\lambda_1},\ldots,c^{\lambda_e}\right)
\end{aligned}$$

which is consistent with $\tilde{\mathbf{c}}_A = m^{\tilde{\mathbf{s}}_A}$, is statistical indistinguishable from partial function evaluations generated by the scheme.

□

Hence, the following theorem is proven.

Theorem 9 *If for a constant number of players the access structure* Γ *is computed by an extended span program over some extension ring of* $\mathbb{Z}$*, then there exists an efficient statistical zero-knowledge and perfect minimal-knowledge RSA function sharing scheme for* Γ*.*

6.5 Robust RSA Function Sharing

We will now present a robust RSA function sharing scheme which is based on the robust secret sharing scheme introduced in the last chapter. Let R, E, H and $(R, M, \mathbf{e}_1)$ be as in the previous sections.

Scheme 8

Distribution phase. Let $s \in E$ be the secret exponent. The dealer shares s as in scheme 4.

Evaluation phase. Let $A = B \cup C$ be the set of players who want to evaluate the secret function at some input $m \in H$ such that B, the honest players, and C, the corrupted players, fulfil (5.1). Every player $i \in A$ computes and broadcasts $R_i = m^{U_i}$. The first

column of R_i is the actual partial function evaluation $\mathbf{c}_i$. Let $\tilde{R}_i$ be what player i claims to be R_i. Analogue to scheme 4 every player computes for all $i \in A$ $D_i = \{j \in A \mid {}^{M_j}R_i^t \neq ({}^{M_i}R_j^t)^t\}$ and $D = \{i \in A \mid D_i \notin \Gamma\}$.
The function evaluation $c = m^s$ can now be computed as in scheme 4 using the partial function evaluations $\mathbf{c}_i$ with $i \in D$.

Proposition 6.6 *The set D computed in the fault detection phase of scheme 8 contains all the honest players but no corrupted player who gave a wrong partial function evaluation $\mathbf{c}_i$.*

Note that ${}^{M_j}R_i^t = {}^{M_j}(m^{U_i^t}) = m^{M_j U_i^t} = m^{(M_i U_j^t)^t} = ({}^{M_i}R_j^t)^t$. So the proof of this proposition goes analogue to the proof of proposition 5.1. The crucial point here is that if $\tilde{R}_i$ differs in the first column from R_i, then ${}^{M_j}\tilde{R}_i^t = ({}^{M_i}R_j^t)^t$ holds at most for an unauthorized set of players j. This follows straight from

Lemma 6.7 *Let S and T be equal dimensional matrices with entries in H^m and different first rows $\mathbf{s}$ and $\mathbf{t}$. Then ${}^{M_j}S = {}^{M_j}T$ holds at most for an unauthorized set of players j.*

Proof. Let ${}^{M_D}S = {}^{M_D}T$ for a set $D \in \Gamma$, i. e. $\mathbf{e}_1$ is of the form $\mathbf{e}_1 = M_D^t \mathbf{v}_D$. Therefore, if the fraction stands for component-wise division, then

$$\frac{\mathbf{s}}{\mathbf{t}} = \frac{{}^{\mathbf{e}_1}S}{{}^{\mathbf{e}_1}T} = \frac{{}^{(M_D^t \mathbf{v}_D)^t}S}{{}^{(M_D^t \mathbf{v}_D)^t}T} = \frac{{}^{\mathbf{v}_D{}^t}({}^{M_D}S)}{{}^{\mathbf{v}_D{}^t}({}^{M_D}T)} = {}^{\mathbf{v}_D{}^t}\left(\frac{{}^{M_D}S}{{}^{M_D}T}\right) = {}^{\mathbf{v}_D{}^t}(1) = \mathbf{1}$$

where (1) stands for the 1-matrix and $\mathbf{1}$ for the 1-vector. But this is a contradiction to $\mathbf{s} \neq \mathbf{t}$.

$\square$

Analogue to the case with no active adversaries, scheme 8 can be modified to a scheme which is statistical zero-knowledge and perfect minimal-knowledge, assumed that the number of players is constant.
Therefore, we finally have

Theorem 10 *If for a constant number of players the access structure Γ is computed by an extended span program over some extension ring of $\mathbb{Z}$, then there exists an efficient, statistical zero-knowledge, perfect minimal-knowledge and verifiable RSA function sharing scheme for Γ.*

If we use the verifiable secret sharing scheme 5 instead of scheme 4 to share the secret exponent in scheme 8 and add two sub protocols, the first one verifying that the modulus n was chosen properly and the second that the secret s really is the inverse of the public key (modulo $\varphi(n)$), then we even get a verifiable RSA function sharing scheme. The correctness of the modulus and the secret key could be verified in the following way. The dealer publishes a large number of modulus, the players choose one at random and the dealer shows that all the others were chosen properly by revealing the corresponding prime numbers. After this and after the players convinced themselves that the shares really define some secret, they can sign some messages chosen at random and verify the signatures with the public key. This guarantees that a faulty dealer will be detected with great probability.

Chapter 7

Conclusion

After proving that for every efficient finite R-module E, where R is a commutative ring with 1, and for every threshold t, there exists an efficient, perfect and homomorphic (t, l)-threshold scheme, we introduced extended span programs, a generalization of the span programs defined by Karchmer and Wigderson in [14], and showed how extended span programs give rise for secret sharing schemes over modules for general access structures. We proved that if in the ring R linear equation systems can be solved, which is, for instance, the case for the familiar rings $\mathbb{Z}$, $\mathbb{Z}_n$, $\mathbb{Z}[X]$ and $\mathbb{Z}[X]/{<}f(X){>}$, then for every access structure which is computed by an efficient extended span program there exists also an efficient, perfect and homomorphic secret sharing scheme over E. Further, we showed how this scheme can be made verifiable, i. e. secure against a faulty dealer and faulty players. Hence, we showed that the three improvements to the Shamir-scheme we mentioned in the introduction, namely

- general access structures,
- security against active adversaries and
- more general secret-space, namely modules instead of fields,

which every one on its own has already been achieved, can be unified in one scheme. As an application, we presented a statistical zero-knowledge and minimal-knowledge verifiable RSA-function sharing scheme. A question we have not answered is the following. If we allow an active adversary an exponentially small success probability, can we weaken the lower bound for the honest players? Another open problem is how far the efficiency of the schemes presented can be improved. We also showed how such an extended span program can be constructed, given a monotone circuit consisting of threshold gates that recognizes the access structure, such that the size of the span program is about the size of the circuit. A question that occurs herewith is for which methods to describe an access structure there exists (and can be computed) an extended span program of the size of the description of the corresponding span program.

Acknowledgments

We would like to thank Ronald Cramer for his encouragement and all the instructive discussions regarding this research. Further, we thank Ivan Damgård for his remarks and suggestions of improvement and Victor Shoup for his idea about how to solve a linear equation system over $\mathbb{Z}$.

Appendix A

Linear Equation Systems over Rings

In the chapters 4 and 5 we assumed that in the commutative ring R with 1 a linear equation system $A\mathbf{x} = b$ can be solved efficiently, at least if there exists a solution. We will now have a closer look at this problem and show that such an equation system can be solved for instance in the rings $\mathbb{Z}$, $\mathbb{Z}_n$, $\mathbb{Z}[X]$ and $\mathbb{Z}[X]/\langle f(X)\rangle$.

First, we have to specify what exactly we mean by solving an equation system. For our purpose it would be enough to demand that one solution can be computed. Nevertheless, we say that in the commutative ring R with 1 linear equation systems $A\mathbf{x} = b$ with $A \in R^{d\times e}$ and $b \in R^d$ can be *solved*, if there exists an algorithm that takes A and b as input and, in the case there exists a solution, computes vectors $\mathbf{x}_0, \mathbf{v}_1, \ldots, \mathbf{v}_r$ such that $\mathbf{x}$ solves $A\mathbf{x} = b$ if and only if $\mathbf{x} \in \{\mathbf{x}_0 + \lambda_1\mathbf{v}_1 + \cdots + \lambda_r\mathbf{v}_r \mid \lambda_1, \ldots, \lambda_r \in R\}$. If there exists no solution to the equation system, the algorithm must be able to detect this. We do not worry too much about efficiency.
Before we investigate in what kind of rings linear equation systems $A\mathbf{x} = \mathbf{b}$ can be solved, we prove

Proposition A.1 *If linear equation systems can be solved in the ring R, then they can also be solved in the rings $R[X]$ and $R/\mathfrak{a}$ for all finitely generated ideals $\mathfrak{a}$ in R.*

Proof. Let $A\mathbf{x} = \mathbf{b}$ be a linear equation system over $R[X]$. By comparing the terms with the same power of X, we get a linear equation system over R which is equivalent to $A\mathbf{x} = \mathbf{b}$. Now consider the equation system $A\mathbf{x} \equiv \mathbf{b} \pmod{\mathfrak{a}}$ with $A \in R^{d\times e}$, $b \in R^d$ and $\mathfrak{a} = <\lambda_1, \ldots, \lambda_m> \subset R$. Then the equation system $A\mathbf{x} \equiv \mathbf{b} \pmod{\mathfrak{a}}$ can be solved by computing $\mathbf{x}, \mathbf{y}_1, \ldots, \mathbf{y}_m$ with $A\mathbf{x} = \mathbf{b} + \lambda_1\mathbf{y}_1 + \ldots + \lambda_m\mathbf{y}_m$. But this is a linear equation system over R and therefore can be solved.

□

We first show that linear equation systems can be solved if (and only if) linear equations $\langle \mathbf{a}, \mathbf{x}\rangle = b$ can be solved, in the same sense.

Proposition A.2 *If there exists an algorithm that solves linear equations $\langle \mathbf{a}, \mathbf{x}\rangle = b$, $\mathbf{a} \in R^e$, then there also exists an algorithm that solves linear equation systems $A\mathbf{x} = \mathbf{b}$, $A \in R^{d\times e}$.*

Proof. By induction on d, the number of rows of A, we show that $A\mathbf{x} = \mathbf{b}$ can be solved if there exists at least one solution. There is nothing to prove for $d = 1$. So let now $d > 1$. Further, let $\mathbf{a}$ and b be the first row of A and the first coordinate of $\mathbf{b}$ and A' and $\mathbf{b}'$ the collection of the other $d-1$ rows and coordinates, respectively, i. e. $A = \binom{\mathbf{a}}{A'}$ and $\mathbf{b} = \binom{b}{\mathbf{b}'}$. By assumption, a vector $\mathbf{x}_0 \in R^e$ and a matrix $V \in R^{e\times r}$ can be computed such that $\mathbf{x} = \mathbf{x}_0 + V\mathbf{y}$, $\mathbf{y} \in R^r$, are exactly the solutions of $\langle \mathbf{a}, \mathbf{x}\rangle = b$, the first row of the equation system. Note that this first row must have a solution as the whole system has one. The solution $\mathbf{x}$ also fulfils the other rows if and only if $\mathbf{b}' = A'\mathbf{x} = A'(\mathbf{x}_0 + V\mathbf{y}) = A'\mathbf{x}_0 + A'V\mathbf{y}$, i. e. if (and only if) $\mathbf{y}$ solves the equation system $A'V\mathbf{y} = \mathbf{b}' - A'\mathbf{x}_0$. Recursively, we can compute a vector $\mathbf{y}_0 \in R^r$ and a matrix $W \in R^{r\times s}$, such that $\mathbf{y} = \mathbf{y}_0 + W\mathbf{z}$, $\mathbf{z} \in R^s$, are the solutions of $A'V\mathbf{y} = \mathbf{b}' - A'\mathbf{x}_0$. It now follows that the vectors $\mathbf{x} = \mathbf{x}_0 + V\mathbf{y} = \mathbf{x}_0 + V(\mathbf{y}_0 + W\mathbf{z}) = (\mathbf{x}_0 + V\mathbf{y}_0) + VW\mathbf{z}$, $\mathbf{z} \in R^s$, are the solutions of $A\mathbf{x} = \mathbf{b}$.
It is clear that if $A\mathbf{x} = \mathbf{b}$ has no solution, then this is detected.

□

An *Euclidean domain* is an integral domain R which has the following two properties.

$$\exists\, N : R \setminus \{0\} \to \mathbb{N} : N(ab) \leq N(a) \forall a, b \in R$$
$$\forall a, b \in R, b \neq 0 \,\exists\, q, r \in R : a = bq + r,\ r = 0 \text{ or } N(r) < N(b)$$

It is known that an Euclidean domain is a principal ideal domain and that the extended Euclidean algorithm can be used to compute, for given $a, b \in R$, elements $s, t \in R$ such that $sa + tb = \gcd(a, b)$. By applying the Euclidean algorithm $n-1$ times it is even possible, for given $a_1, \dots, a_n \in R$, to compute $s_1, \dots, s_n \in R$ such that $s_1 a_1 + \cdots + s_n a_n = \gcd(a_1, \dots, a_n)$, i. e. to compute one solution of a linear equation $\langle \mathbf{a}, \mathbf{x} \rangle = b$ where b is a multiple of $\gcd(a_1, \dots, a_n)$. We will in the following show that all solutions can be computed. First, we look at homogene equations.

Proposition A.3 *Let R be an Euclidean domain. Then homogene, linear equations $\langle \mathbf{a}, \mathbf{x} \rangle = 0$, $\mathbf{a} \in R^e$, can be solved.*

Proof. We may wlog assume that no coordinate of $\mathbf{a} = (a_1, \dots, a_e)$ is zero and that $\gcd(a_1, \dots, a_e) = 1$. The case $e = 1$ is trivial. Let now $e > 1$. Set $d = \gcd(a_2, \dots, a_e)$ and $\mathbf{v}_1 = \binom{d}{\mathbf{v}_1'}$, where the $\mathbf{v}_1'$ fulfils $\langle \mathbf{a}', \mathbf{v}_1' \rangle = -da_1$ for $\mathbf{a}' = (a_2, \dots, a_e)$. $\mathbf{v}_1'$ can be computed with the extended Euclidean algorithm. Further, for $i = 2, \dots, e-1$, let $\mathbf{v}_i = \binom{0}{\mathbf{v}_i'}$ where $\mathbf{v}_2', \dots, \mathbf{v}_{e-1}' \in R^{e-1}$, recursively computed, span the solution space of $\langle \mathbf{a}', \mathbf{x}' \rangle = 0$. It is clear that $\mathbf{v}_1, \cdots, \mathbf{v}_{e-1}$ are solutions of $\langle \mathbf{a}, \mathbf{x} \rangle = 0$, we will now show that they span the whole solution space $\mathcal{V}$. Let $\mathcal{V}$ be spanned by the vectors $\mathbf{w}_1, \mathbf{w}_2, \dots$. It is easy to see that for $i = 1, 2, \dots$ the first coordinate of $\mathbf{w}_i$, say ω_i, must be a multiple of d. Therefore, $\frac{\omega_i}{d}$ is an integer and the vectors $\mathbf{v}_1, \mathbf{w}_1 - \frac{\omega_1}{d}\mathbf{v}_1, \mathbf{w}_2 - \frac{\omega_2}{d}\mathbf{v}_1, \dots$ also span $\mathcal{V}$. But $\mathbf{w}_i - \frac{\omega_i}{d}\mathbf{v}_1$ is of the form $\mathbf{w}_i - \frac{\omega_i}{d}\mathbf{v}_1 = \binom{0}{\mathbf{w}_i'}$ where $\mathbf{w}_i'$ solves $\langle \mathbf{a}', \mathbf{x}' \rangle = 0$ and therefore $\mathbf{w}_i - \frac{\omega_i}{d}\mathbf{v}_1$ can be written as a linear combination of the $\mathbf{v}_2, \dots, \mathbf{v}_{e-1}$, which proves that the vectors $\mathbf{v}_1, \dots, \mathbf{v}_{e-1}$ span the whole solution space $\mathcal{V}$.

□

Proposition A.4 *Let R be an Euclidean domain. Then linear equations $\langle \mathbf{a}, \mathbf{x} \rangle = b$, $\mathbf{a} \in R^e$, can be solved.*

Proof. We may wlog assume that no coordinate of $\mathbf{a}$ is zero. There exists a solution if and only if b is a multiple of the greatest common divisor of the coordinates of $\mathbf{a}$. Therefore a solution $\mathbf{x}_0$ of $\langle \mathbf{a}, \mathbf{x} \rangle = b$ can be computed using the extended Euclidean algorithm. The solution space is now given by $\mathbf{x}_0 + \lambda_1 \mathbf{v}_1 + \cdots + \lambda_{e-1}\mathbf{v}_{e-1}$ where the vectors $\mathbf{v}_1, \cdots, \mathbf{v}_{e-1} \in R^e$, computed according to the previous proposition, span the solution space of the homogene equation $\langle \mathbf{a}, \mathbf{x} \rangle = 0$.

□

¿From this and from proposition A.1 it follows

Corollary A.5 *In the rings $\mathbb{Z}$, $\mathbb{Z}_n$, $\mathbb{Z}[X]$ and $\mathbb{Z}[X]/{<}f(X){>}$ linear equation systems can be solved.*

The problem is that if we solve a linear equation system $A\mathbf{x} = \mathbf{b}$ over $R = \mathbb{Z}$ using the method described in the proof of proposition A.2, then the numbers involved might grow exponentially.
A more efficient method would be the following, using the L^3 algorithm to compute the kernel of a matrix with integer entries [7, p. 98]. Note that $A\mathbf{x} = \mathbf{b}$ is equivalent to $(\mathbf{b}, A)\binom{x_0}{\mathbf{x}} = \mathbf{0}$, $x_0 = -1$. Using the L^3 algorithm, we get a matrix V such that $\mathrm{Ker}\,(\mathbf{b}, A) = \{V\mathbf{y} \mid \mathbf{y} \in \mathbb{Z}^s\}$. If we further compute $\mathbf{y}_0$ and W such that $\{\mathbf{y}_0 + W\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^{s-1}\}$ is the solution space of $\langle \mathbf{v}, \mathbf{y} \rangle = -1$, where $\mathbf{v}$ is the first row of V, then $\{V\mathbf{y}_0 + VW\mathbf{z} \mid \mathbf{z} \in \mathbb{Z}^{s-1}\}$ is the solution space of $(\mathbf{b}, A)\binom{x_0}{\mathbf{x}} = \mathbf{0}$, $x_0 = 1$.
In this context, it would be interesting to find out if the L^3 algorithm also works for other rings, for instance for other Euclidean domains.

Bibliography

[1] M. Ben-Or, S. Goldwasser, A. Wigderson: *Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract).* In Proc. of the 20th Annual ACM Symp. on Theory of Computing, pp. 1–10, Chicago, Illinois, 2–4 May 1988.

[2] J. Benaloh, J. Leichter: *Generalized Secret Sharing and Monotone Functions.* In S. Goldwasser, editor, Advances in Cryptology – CRYPTO '88, volume 403 of Lecture Notes in Computer Science, pp. 27–35, 21–25 August 1988. Springer-Verlag, 1990.

[3] E. F. Brickell: *Some Ideal Secret Sharing Schemes.* Journal of Combinatorial Mathematics and Combinatorial Computing, **9** (1989), pp. 105–113.

[4] D. Chaum, C. Crépeau, I. Damgård: *Multiparty Unconditionally Secure Protocols (extended abstract).* In Proc. of the 20th Annual ACM Symp. on the Theory of Computing, pp. 11-19, Chicago, Illinois, 2–4 May 1988.

[5] D. Chaum, H. Van Antwerpen: *Undeniable Signatures.* In G. Brassard, editor, Advances in Cryptology – CRYPTO '89, volume 435 of Lecture Notes in Computer Science, pp. 212–216, 20–24 August 1989. Springer-Verlag, 1990.

[6] B. Chor, S. Goldwasser, S. Micali, B. Awerbuch. *Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (extended abstract).* In 26th Annual Symp. on Foundations of Computer Science, pp. 383-395, Portland, Oregon, 21–23 October 1985. IEEE.

[7] H. Cohen: *A Course in Computational Algebraic Number Theory* Springer-Verlag, Berlin, Heidelberg, 1993.

[8] R. Cramer, I. Damgård, U. Maurer: *Span Programs and General Secure Multi-Party Computation.* BRICS Report Series RS-97-28, available from `http://www.brics.dk`, 1997.

[9] A. De Santis, Y. Desmedt, Y. Frankel, M. Yung: *How to Share a Function Securely (extended summary).* In Proc. of the 26th Annual ACM Symp. on the Theory of Computing, pp. 522–533, Montréal, Québec, Canada, 23–25 May 1994.

[10] Y. G. Desmedt, Y. Frankel: *Homomorphic Zero-Knowledge Threshold Schemes over any Finite Abelian Group.* SIAM J. DISC. MATH., Vol. 7, No. 4, pp. 667–679, November 1994.

[11] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin: *Robust and Efficient Sharing of RSA Functions.* In N. Koblitz, editor, Advances in Cryptology – CRYPTO '96, volume 1109 of Lecture Notes in Computer Science, pp. 157–172, 18–22 August 1996. Springer-Verlag.

[12] M. Hirt, U. Maurer: *Complete Characterization of Adversaries Tolerable in Secure Multiparty Computation (extended abstract).* In Proc. of the 16th Annual ACM Symp. on Principles of Distributed Computing, pp. 25–34, Santa Barbara, California, 21–24 August 1997.

[13] M. Ito, A. Saito, T. Nishizeki: *Secret Sharing Scheme Realizing General Access Structures.* Proc. IEEE Globecom '87, pp. 99–102, 1987.

[14] M. Karchmer, A. Wigderson: *On Span Programs.* In Proc. of the 8th Annual Structure in Complexity Theory Conference, pp. 102–111, San Diego, California, 18–21 May 1993. IEEE Computer Society Press.

[15] S. Lang: *Algebra, 3rd ed.* Addison-Wesley Publishing Company, Jan. 1997.

[16] M. Marcus: *Introduction to Modern Algebra.* Marcel Dekker, New York, 1978.

[17] D. S. Mitrinović, J. Sándor, B. Crstici: *Handbook of Number Theory* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[18] T. Rabin: *Robust Sharing of Secrets when the Dealer is Honest or Faulty.* Journal of the ACM, 41(6):1089-1109, November 1994

[19] T. Rabin, M. Ben-Or: *Verifiable Secret Sharing and Multiparty Protocols with Honest Majority (extended abstract).* In Proc. of the 21st Annual ACM Symp. on the Theory of Computing, pp. 73–85, Seattle, Washington, 15–17 May 1989.

[20] R. L. Rivest, A. Shamir, L. Adleman: *A Method for Obtaining Digital Signatures and Public Key Cryptosystems.* Comm. of the AMC, 21 (1978), pp. 120–126.

[21] A. Shamir: *How to Share a Secret.* Comm. of the ACM, 22 (1979), pp. 612–613.