Generalized Number Systems and Secure Electronic Elections

Ph.D. Thesis

Andrea Huszti

Supervisor: Prof. Attila Pethő

University of Debrecen
Doctoral Committee of Natural Sciences
Doctoral School of Computer Sciences

Debrecen, 2008

Ezen értekezést a Debreceni Egyetem Természettudományi Tudományterületi Doktori Tanács Informatikai Tudományok Doktori Iskola Digitális kommunikáció programja keretében készítettem a Debreceni Egyetem doktori (PhD) fokozatának elnyerése céljából.
Debrecen, 2008. december 15.

.....
Huszti Andrea
jelölt

Tanúsítom, hogy Huszti Andrea doktorjelölt 2004 - 2007 között az Informatikai Tudományok Doktori Iskola Digitális kommunikáció programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Az értekezés elfogadását javasolom.

Debrecen, 2008. december 15.

.....
Dr. Pethő Attila
témavezető

Generalized Number Systems and Secure Electronic Elections

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében az informatika tudományágban.

Írta: Huszti Andrea okleveles matematika, ábrázoló geometria és informatika tanár.

Készült a Debreceni Egyetem Informatikai Tudományok Doktori Iskolája (Digitális kommunikáció programja) keretében.

Témavezető: Dr. Pethő Attila

A doktori szigorlati bizottság:

elnök: Dr. ....

tagok: Dr. ....

Dr. ....

A doktori szigorlat időpontja: 200... ....

Az értekezés bírálói:

Dr. ....

Dr. ....

Dr. ....

A bírálóbizottság:

elnök: Dr. ....

tagok: Dr. ....

Dr. ....

Dr. ....

Dr. ....

Az értekezés védésének időpontja: 200... ....

Contents

Chapter 1

Introduction

1.1 Historical background

The present dissertation is based on two more or less independent topics, dealing with generalized number systems and cryptographically secure electronic elections. In the first part we investigate canonical number systems in quartic algebraic number fields, then we characterize three-dimensional symmetric shift radix systems. In the second part of the dissertation two secure election schemes are described, one of them is based on blind signatures the other one uses homomorphic encryptions. Both schemes are secure and designed to be implemented in a practical environment, hence they do not employ an untappable channel or voting booth.

Canonical number systems can be viewed as natural generalizations of radix representations of ordinary integers (Grünwald [34]) to algebraic integers. An example of a canonical number system was first studied by Knuth [50],[51]. They showed that the complex number $b = -1 + \sqrt{-1}$ can be used as a base for a number system which admits finite representations for each Gaussian integer. This means that each nonzero $\gamma \in \mathbb{Z}[\sqrt{-1}]$ has a unique representation of the shape

$$\gamma = c_0 + c_1 b + \cdots + c_h b^h$$

with $c_i \in \{0,1\}$ $(0 \le i \le h)$, $c_h = 1$ and $h \in \mathbb{N}$.

This observation has been generalized and studied extensively in the last decades. Gilbert, Kátai, Kovács and Szabó ([30], [43], [44], [45]) extended Knuth's notion to arbitrary quadratic number fields. For each quadratic number field K with maximal order $\mathbb{Z}_K$ they have characterized all the elements $b \in \mathbb{Z}_K$ which can serve as a base of a number system in $\mathbb{Z}_K$. Let α be an algebraic integer then b is called a base for $\mathbb{Z}[\alpha]$ if each non-zero $\gamma \in \mathbb{Z}[\alpha]$ admits a unique representation of the shape

$$\gamma = c_0 + c_1 b + \cdots + c_h b^h$$

with $c_i \in \{0, 1, \ldots, |N(b) - 1|\}$ and $c_h \neq 0$ ($N(\cdot)$ is the norm of b over $\mathbb{Q}$).

Later, Kovács and Pethő [49] (see also [47]) extended this notion to arbitrary number fields and gave some partial results on the characterization of the bases.

Various variants of canonical number systems have been studied in the literature. Kovács [48] studied number systems with integer digits in rings. Pethő [66] considered simultaneous representation of several elements. Scheicher and Thuswaldner [77] investigated number systems in polynomial rings over finite field (cf. also [49]).

CNS have connections to the theories of finite automata (see e.g. K. Scheicher [75], J. M. Thuswaldner [86]) and fractal tilings (see e.g. S. Akiyama and J. M. Thuswaldner [10]). S. Akiyama et al. [2] put canonical number systems (CNS) into a more general framework thereby opening links to other areas, e.g. to a long-standing problem on Salem numbers.

In [2] a dynamical system called shift radix system (SRS) has been introduced. SRS are related to number systems as β-expansions (*cf.* for instance [27, 62, 70]) or canonical number systems. Indeed they form a unification and generalization of these notions of number systems. More details about SRS and their relation to β-expansions and CNS can be found in [2], [3], [83].

CNS bases in quadratic and cubic fields were characterized by several authors (see [43],[44],[30],[33],[86],[8],[49],[5]). Pethő described CNS bases in a class of biquadratic number fields in [67]. In this dissertation CNS

bases in quartic number fields are characterized, including cyclotomic and simplest quartic fields. We deal with an important variant of SRS, the so-called symmetric shift radix systems (SSRS), which was introduced in [9]. SSRS analogously to SRS are related to symmetric β-expansions and symmetric canonical number systems. In [9] two dimensional SSRS were studied, we extend investigations to the three dimensional case.

Cryptographic protocols, for example secure voting schemes, are as strongly related to number theory as generalized number systems. Security of constructions of cryptographic primitives are based on problems from number theory which seem to be computationally intractable. The most well-known of these problems are calculating discrete logarithms and factoring composite integers.

The research on electronic voting is a very important topic for the progress of democracy. If a secure and convenient electronic voting system is provided, it will be used more frequently to collect people's opinion for many kinds of political and social decisions through cyber space. Traditional paper-based voting can be time consuming and inconvenient. Electronic voting not only accelerates the whole process, but makes it less expensive and more comfortable for the voters and the authorities as well. It also reduces the chances of errors.

Electronic election schemes according to the applied cryptographic techniques can be categorized into three main models.

The mix-net model. Chaum [18] introduces the concept of a mix-net that is built up from several linked servers called mixes. Each mix randomizes input messages and outputs the permutation of them, such that the input and output messages are not linkable to each other. Several schemes based on mix-nets are proposed in the literature ([61],[74],[42]).

The blind signatures model. The concept of blind signatures was introduced by Chaum [19]. During the Authorizing stage a voting authority authenticates a token, (usually an encrypted vote) without knowing the contents. This way of authentication is achieved by applying blind signatures. Even if later the (un-blinded) signature is made public, it is impossible to connect the signature to the signing process, *i.e.* to the voter. Schemes

based on blind signatures usually use anonymous channel, during the Voting stage voters send the un-blinded signature and the encryption of the ballot to a voting authority, assuring the anonymity of the sender. Tallying stage consists of two phases, *opening* and *counting* phases. During opening phase encrypted ballots are decrypted and in the counting phase only valid votes are collected. The result of the election is made public. For further schemes see [28], [38], [58], [59], [69].

The homomorphic encryption model. Schemes based on homomorphic encryptions employ s authorities in order to manage Voting and Tallying stages. These schemes use secret sharing scheme either to share the decryption key, or to share the vote itself. The following two alternatives appear in the literature:

- A voter creates v_i, $i = 1, \ldots, s$ shares of his secret vote, and sends encrypted share $E(v_i)$ to A_i. A_i collects all ith encrypted shares, in Tallying stage decrypts the sum of the received shares. At the end authorities together calculate the result of the election.

- Decryption key, used at the end of the whole procedure in order to decrypt the sum of the votes, is shared among the s authorities.

For secret sharing Shamir's secret sharing system can be applied, where there are at most $t < s$ malicious authorities, meaning, that at least $t + 1$ authorities are necessary to generate the secret information.

Since the election result is computed by decrypting the product of encrypted votes, the encrypted vote itself is never decrypted. It is essential to prove that the encrypted vote is formed properly, that the encrypted value is really the encryption of one of the candidates. This is proved with a use of zero-knowledge proof, so without obtaining any knowledge about the vote itself. An encrypted vote is valid if the corresponding zero-knowledge proof runs correctly.

Models based on homomorphic encryption are [22], [52] and [36]. Most of the voting systems use ElGamal encryptions, alternative homomorphic encryption is Pallier cryptosystem [60], schemes based on it are proposed cf. [11], [25].

The concept of *receipt-freeness* and *uncoercibility* were introduced by Benaloh and Tuinstra [12]. Roughly speaking, receipt-freeness is the inability of a voter to prove an adversary that he voted in a particular manner, even if the voter wishes to do so. For formal definition we refer to [59]. Several receipt-free and uncoercible voting schemes are designed with applying untappable channels or voting booths, that are unpractical [59]. Another solution in order to achieve receipt-freeness is that the scheme employs an extra tamper-resistant hardware [54]. The two secure electronic election schemes presented here can be implemented in practice, they do not require untappable channels and voting booths. Extra hardware is not employed, either.

In the next section we give a detailed presentation overview of our results.

1.2 Presentation overview and our results

The present work consists of two main topics, these topics lead into two more or less independent directions. Chapter one and two deal with generalized number systems, chapter three and four are contain cryptographically secure electronic elections. The appendix (Chapter 6) details the Proof of Lemma 2.4.9. More precisely, this dissertation consists of the following parts.

The introduction (first chapter) contains the historical background, the presentation overview and our main results.

In the second chapter we deal with Canonical Number Systems. Our main result is the characterization of CNS bases in algebraic number fields including quartic cyclotomic fields, simplest quartic fields and two families of orders in quartic number fields. By a theorem of B. Kovács [47] there exists CNS in an order if and only if there exists power integral bases. For finding CNS bases a modified version of the algorithm given by B. Kovács and A. Pethő [49] is applied. This algorithm assumes existence of a set, that

contains representatives of the equivalence classes of generators of power integral bases of the given order.

Chapter three is devoted to three-dimensional Symmetric Shift Radix Systems. Let us denote

$$\begin{aligned} \mathcal{D}_d^0 &:= \left\{ \mathbf{r} \in \mathbb{R}^d \,|\tau_{\mathbf{r}} \text{ is an SSRS} \right\} \\ \mathcal{D}_d &:= \left\{ \mathbf{r} \in \mathbb{R}^d \,|\tau_{\mathbf{r}} \text{ is eventually periodic} \right\}. \end{aligned}$$

As a new result we prove that $\mathcal{D}_3^0$ is an union of four polyhedra and a polygon, by employing the algorithm that is established for SSRS in [9]. This algorithm constructs a finite directed graph and finds all nonzero primitive cycles in it. Roughly speaking these cycles give periods which generate cutout polyhedra and we get $\mathcal{D}_3^0$ by subtracting the generated cutout polyhedra from $\mathcal{D}_3$.

Chapter four presents the building blocks of our election protocols and all communication channels usually employed in voting are enumerated.

In chapter five after describing requirements and participants of voting schemes two new secure election protocols are detailed. Both of them possess all basic requirements and can be implemented in practice.

In section 5.3 the scheme is based on blind signatures, requires only two authorities (Registry and Voting Authority) and does not employ complex primitives like zero-knowledge proofs or threshold cryptosystems. Our election scheme satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability and coercion-resistance. It can be implemented in practice, since it does not apply impractical untappable channels or voting booths. It consists of three distinctive stages: *Authorizing*, *Voting* and *Tallying*. During the Voting stage voters create their ballots. Ballots contain the selected candidate and blind signature is applied to hide it from the Voting Authority. This scheme is offered to be employed in an environment, where authorities participating do not collude and the Voting Authority does not collaborate with adversaries.

In 5.4 our protocol is based on homomorphic encryptions, it assumes existence of several authorities and it uses distributed ElGamal encryption [63]. This scheme is based on [22] that is not possessing the property of receipt-freeness or uncoercibility. There are two models based on [22] that are designed to be receipt-free in the literature: [52] and [36]. First one applies an honest verifier, the second one uses an untappable channel. Our scheme does not employ voting booths or untappable channels, it requires an anonymous return channel, hence it can be implemented in practice. We do not have an honest verifier, either. The only assumption is that among the Voting Authorities participating in distributed key generation and decryption there is at least one authority that is honest. The scheme satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability, receipt-freeness, uncoercibility and protects against randomization and forced-abstention attacks.

Chapter six contains the Proof of Lemma 2.4.9. It is quite complicated, therefore it can be found in the appendix.

1.3 Credits

Results of *Canonical Number Systems* are based on

> H. Brunotte, A. Huszti, A. Pethő, *Bases of canonical number systems in quartic algebraic number fields*, Journal de Théorie des Nombres de Bordeaux, **18** (2006), 537 – 559.

Results of *Symmetric Shift Radix Systems* are based on

> A. Huszti, K. Scheicher, P. Surer, J. M. Thuswaldner, *Three-dimensional symmetric shift radix systems*, Acta Arithmetica, **129** (2007), 147 – 166.

Results of *A Coercion-Resistant Voting Scheme Based on Blind Signatures* are based on

A. Huszti, *A Secure Electronic Voting Scheme*, Periodica Polytechnica Electrical Engineering, **51/3-4** (2007), 1 – 6.

Results of *A Receipt-Free Homomorphic Election Scheme* are based on

A. Huszti, *A Homomorphic Encryption-Based Secure Electronic Voting Scheme*, submitted for publication.

Chapter 2

Canonical Number Systems

In this chapter after we define canonical number systems, shift radix systems and give their basic properties, we present a slightly modified version of the algorithm established by B. Kovács and A. Pethő [49] for the determination of CNS bases of orders of algebraic number fields. (See also [67] for a comprehensive description of the original algorithm and its background.) CNS bases are explicitly known for some quadratic, cubic and quartic fields ([43],[44],[30],[33],[86],[8], [49], [5], [67]). The list of CNS bases of simplest cubic fields given in [5] is also extended here. The modified algorithm is exploited for some families of number fields of low degrees; our main applications are cyclotomic and simple fields of degree four.

The results of this chapter are contained in our paper [17]. This paper is a joint work with Horst Brunotte and Attila Pethő.

2.1 CNS bases of algebraic number fields

The investigation of the question wether an algebraic number field is monogenic is a classical problem in algebraic number theory (cf. [29]). According to B. Kovács [47] the existence of a power integral basis in an algebraic number field is equivalent to the existence of a canonical number system for its maximal order. Moreover, using a deep result of K. Győry [35] on gener-

ators of orders of algebraic number fields B. Kovács [47] proved that up to translation by integers there exist only finitely many canonical number systems in the maximal order of an algebraic number field.

In the sequel we denote by $\mathbb{Q}$ the field of rational numbers, by $\mathbb{Z}$ the set of integers and by $\mathbb{N}$ the set of nonnegative integers. For an algebraic integer γ we let $\mu_\gamma \in \mathbb{Z}[X]$ be its minimal polynomial and $\mathcal{C}_\gamma$ the set of all CNS bases for $\mathbb{Z}[\gamma]$.

2.1.1 Definition. Let

$$P(X) = X^d + p_{d-1}X^{d-1} + \cdots + p_1X + p_0 \in \mathbb{Z}[X],\ N = \{0, 1, \ldots, |p_0| - 1\}$$

and $\mathcal{R} := \mathbb{Z}[X]/P(X)\mathbb{Z}[X]$ and denote the image of X under the canonical epimorphism from $\mathbb{Z}[X]$ to $\mathcal{R}$ by x. If every non-zero element $A(x) \in \mathcal{R}$ can be written uniquely in the form

$$A(x) = a_0 + a_1x + \cdots + a_lx^l$$

with $a_0, \ldots, a_l \in N, a_l \neq 0$, we call (P, N) a *canonical number system* (CNS for short). $P(X)$ is called CNS polynomial, to N we refer as the set of digits.

We denote by $\mathcal{C}$ the set of CNS polynomials; for the general definition of CNS polynomials we refer the reader to A. Pethő [65], however, for our purposes it suffices to keep in mind that α is a CNS basis for $\mathbb{Z}[\alpha]$ if and only if μ_α is a CNS polynomial. It can algorithmically be decided whether a given integral polynomial is a CNS polynomial or not (see [1]).

B. Kovács [47] introduced the following set of polynomials

$$\begin{aligned}\mathcal{K} \;=\; & \{p_dX^d + p_{d-1}X^{d-1} + \cdots + p_0 \in \mathbb{Z}[X] \mid \\ & d \geq 1, 1 = p_d \leq p_{d-1} \leq \ldots \leq p_1 \leq p_0 \geq 2\}\end{aligned}$$

which plays a decisive role in the theory of CNS polynomials (see [1], Theorem 2.3).

2.1.2 Lemma. *(B. Kovács – A. Pethő) For every nonzero algebraic integer α the following constants can be computed effectively:*

$$k_\alpha = \min\{k \in \mathbb{Z} \,|\, \mu_\alpha(X+n) \in \mathcal{K} \text{ for all } n \in \mathbb{Z} \text{ with } n \geq k\},$$

$$c_\alpha = \min\{k \in \mathbb{Z} \,|\, \mu_\alpha(X+k) \in \mathcal{C}\}.$$

Proof See [49], Section 5. □

Note that $c_\alpha \leq k_\alpha$ by ([47], Lemma 2) and that if β is a conjugate of α then $k_\beta = k_\alpha$ and $c_\beta = c_\alpha$.

2.1.3 Corollary. *If α is a CNS basis for an order R then $c_\alpha \leq 0, \alpha - c_\alpha$ is a CNS basis for R, but $\alpha - c_\alpha + 1$ is not a CNS basis for R.*

Proof This is clear by the definitions. □

To a polynomial $P(X) = p_d X^d + p_{d-1}X^{d-1} + \cdots + p_0 \in \mathbb{Z}[X], p_d = 1$ we associate the mapping $\tilde{\tau}_P = \tilde{\tau} : \mathbb{Z}^d \to \mathbb{Z}^d$ defined by

$$\tilde{\tau}_P(\mathbf{a}) = \left(a_2, \ldots, a_d, -\left\lfloor \frac{p_1 a_d + \cdots + p_d a_1}{p_0} \right\rfloor\right),$$

where $\mathbf{a} = (a_1, \ldots, a_d) \in \mathbb{Z}^d$. This turned out very useful to prove $P(X) \in \mathcal{C}$. Indeed Brunotte [15] proved the following theorem, that gives an efficient algorithm for testing if a polynomial is CNS or not.

2.1.4 Theorem. *Assume that $E \subseteq \mathbb{Z}^d$ has the following properties:*

(i) $(1, 0, \ldots, 0) \in E$,

(ii) $-E \subseteq E$,

(iii) $\tilde{\tau}(E) \subseteq E$,

(iv) for every $e \in E$ there exist some $l > 0$ with $\tilde{\tau}^l(e) = 0$.

Then $P(X) \in \mathcal{C}$.

2.1.5 Definition. (***cf.*** **[2])** Let $d \geq 1$ be an integer, $\mathbf{r} \in \mathbb{R}^d$, and let

$$\tilde{\tau}_{\mathbf{r}} : \mathbb{Z}^d \to \mathbb{Z}^d, \quad \mathbf{a} = (a_1, \ldots, a_d) \mapsto (a_2, \ldots, a_d, -\lfloor \mathbf{ra} \rfloor),$$

where $\mathbf{ra} = r_1 a_1 + r_2 a_2 + \cdots + r_d a_d$, *i.e.*, the inner product of the vectors $\mathbf{r}$ and $\mathbf{a}$. Then $\tilde{\tau}_{\mathbf{r}}$ is called a *shift radix system* (for short SRS), if

$$\forall \mathbf{a} \in \mathbb{Z}^d \quad \exists n \in \mathbb{N} : \tilde{\tau}_{\mathbf{r}}^n(\mathbf{a}) = \mathbf{0}.$$

Let

$$\tilde{\mathcal{D}}_d := \left\{ \mathbf{r} \in \mathbb{R}^d \,\middle|\, \forall \mathbf{a} \in \mathbb{Z}^d \, \exists n, l \in \mathbb{N} : \tilde{\tau}_{\mathbf{r}}^k(\mathbf{a}) = \tilde{\tau}_{\mathbf{r}}^{k+l}(\mathbf{a}) \; \forall k \geq n \right\} \text{ and}$$

$$\tilde{\mathcal{D}}_d^0 := \left\{ \mathbf{r} \in \mathbb{R}^d \,|\tilde{\tau}_{\mathbf{r}} \text{ is an SRS} \right\},$$

2.1.6 Theorem. *(S. Akiyama et al. [2]) Let* $P(X) = X^d + p_{d-1}X^{d-1} + \cdots + p_1 X + p_0 \in \mathbb{Z}[\mathbb{X}]$. *Then* $P(X) \in \mathcal{C}$ *if and only if* $\mathbf{r} = \left(\frac{1}{p_0}, \frac{p_{d-1}}{p_0}, \ldots, \frac{p_1}{p_0}\right) \in \tilde{\mathcal{D}}_d^0$.

2.1.7 Theorem. *(S. Akiyama et al. [2]) Let* $\mathbf{r}_1, \ldots, \mathbf{r}_k$ *be points of* $\tilde{\mathcal{D}}_d$ *and denote by* H *the convex hull of* $\mathbf{r}_1, \ldots, \mathbf{r}_k$. *We assume that* H *is contained in the interior of* $\tilde{\mathcal{D}}_d$ *and is sufficiently small in diameter. For* $\mathbf{z} \in \mathbb{Z}^d$ *take* $M(\mathbf{z}) = max_{1 \leq i \leq k}\{-\lfloor \mathbf{r}_i \mathbf{z} \rfloor\}$. *Then there exist an algorithm to create a finite directed graph* (V, E) *with vertices* $V \subset \mathbb{Z}^d$ *and edges* $E \in V \times V$ *which satisfy*

1. *each d-dimensional standard unit vector* $(0, \ldots, 0, \pm 1, 0, \ldots, 0) \in V$
2. *for each* $\mathbf{z} = (z_1, \ldots . z_d) \in V$ *and*

   $$j \in [-M(-\mathbf{z}), M(\mathbf{z})] \cap \mathbb{Z}$$

   we have $(z_2, \ldots, z_d, j) \in V$ *and a directed edge* $(z_1, \ldots, z_d) \longrightarrow (z_2, \ldots, z_d, j)$ *in* E.
3. $H \cap \mathcal{D}_d^0 = H \setminus \cup_\pi P(\pi)$, *where* π *are taken over all nonzero primitive cycles of* (V, E)*; here* $P(\pi)$ *denotes a certain convex polyhedron defined by* π.

The following notion seems to be convenient for the intentions of the present note.

2.1.8 Definition. The algebraic integer α is called a *fundamental CNS basis* for R if it satisfies the following properties:

(1) $\alpha - n$ is a CNS basis for R for all $n \in \mathbb{N}$.

(2) $\alpha + 1$ is a not CNS basis for R.

2.1.9 Theorem. *Let γ be an algebraic integer. Then there exist finite effectively computable disjoint subsets $\mathcal{F}_0(\gamma), \mathcal{F}_1(\gamma) \subset \mathcal{C}_\gamma$ with the properties:*

(i) For every $\alpha \in \mathcal{C}_\gamma$ there exists some $n \in \mathbb{N}$ with $\alpha + n \in \mathcal{F}_0(\gamma) \cup \mathcal{F}_1(\gamma)$.

(ii) $\mathcal{F}_1(\gamma)$ consists of fundamental CNS bases for $\mathbb{Z}[\gamma]$.

Proof By ([49], Theorem 5) there exist finitely many effectively computable

$$\alpha_1, \ldots, \alpha_t \in \mathbb{Z}[\gamma], \quad n_1, \ldots, n_t \in \mathbb{Z}, \quad N_1, \ldots, N_t \subset \mathbb{Z}$$

such that for every $\alpha \in \mathbb{Z}[\gamma]$ we have

$$\begin{aligned} &\alpha \in \mathcal{C}_\gamma \iff \alpha = \alpha_i - h, \\ &\text{for } i \in \{1, \ldots, t\}, h \in \mathbb{Z} \text{ and } h \geq n_i \text{ or } h \in N_i. \end{aligned} \tag{2.1.1}$$

Therefore the set

$$F := \{\alpha_i - n_i \,|\, i = 1, \ldots, t\} \cup \bigcup_{i=1}^{t} \{\alpha_i - h \,|\, h \in N_i\}$$

is a finite effectively computable subset of $\mathcal{C}_\gamma$.

For every $\alpha \in F$ let

$$M_\alpha = \{m \in \mathbb{Z} \,|\, m \leq k_\alpha, \alpha - k \in \mathcal{C}_\gamma \text{ for all } k = m, \ldots, k_\alpha\}.$$

Observing $m \geq c_\alpha$ for all $m \in M_\alpha$ we see using Lemma 1. that M_α is a nonempty finite effectively computable set. Let

$$m_\alpha = \min M_\alpha$$

and

$$\mathcal{F}_0(\gamma) = \{\alpha - c_\alpha \,|\, \alpha \in F, m_\alpha > c_\alpha\}, \quad \mathcal{F}_1(\gamma) = \{\alpha - c_\alpha \,|\, \alpha \in F, m_\alpha = c_\alpha\}.$$

We show that $\mathcal{F}_1(\gamma)$ consists of fundamental CNS bases for $\mathbb{Z}[\gamma]$. Let $\varphi \in \mathcal{F}_1(\gamma)$, hence $\varphi = \alpha - c_\alpha$ with some $\alpha \in F$. By Corollary 2.1.3 we have $\varphi \in \mathcal{C}_\gamma, \varphi + 1 \notin \mathcal{C}_\gamma$. For $n \in \mathbb{N}$ we find

$$\varphi - n = \alpha - (m_\alpha + n) \in \mathcal{C}_\gamma,$$

because for $m_\alpha + n \leq k_\alpha$ this is clear by the definition of m_α, and for $m_\alpha + n > k_\alpha$ we have $\mu_{\varphi - n} = \mu_\alpha(X + (m_\alpha + n)) \in \mathcal{K}$ and therefore $\varphi - n \in \mathcal{C}_\gamma$ by ([47], Lemma 2).

Finally, let $\beta \in \mathcal{C}_\gamma$. By (2.1.1) there are $i \in \{1, \ldots, t\}$ and $h \in \mathbb{Z}$ with

$$\beta = \alpha_i - h \text{ and } h \geq n_i \text{ or } h \in N_i.$$

If $h \in N_i$ then $\beta \in F$ and $\beta - c_\beta \in \mathcal{F}_0(\gamma) \cup \mathcal{F}_1(\gamma)$ by Corollary 2.1.3. If $h \geq n_i$ then $\alpha = \alpha_i - n_i \in F, h - n_i - c_\alpha \in \mathbb{N}$ and

$$\beta + (h - n_i - c_\alpha) = \alpha - c_\alpha \in \mathcal{F}_0(\gamma) \cup \mathcal{F}_1(\gamma).$$

□

2.1.10 Remark. Note that $\varphi \in \mathcal{F}_0(\gamma)$ implies $\varphi - n \in \mathcal{F}_1(\gamma)$ for some $n \in \mathbb{N} \setminus \{0\}$. Therefore the theorem of B. Kovács ([47], Lemma 2) can be rephrased in the following form: An algebraic number field is monogenic if and only if there exists a fundamental CNS basis for its maximal order.

Slightly modifying the algorithm of B. Kovács and A. Pethő [49] we now present the algorithm for finding the above mentioned sets $\mathcal{F}_0(\gamma)$ and $\mathcal{F}_1(\gamma)$. The (finite) set T is introduced to keep track of the calculations performed;

(CNS basis computation)

[Input] A nonzero algebraic integer γ and a (finite) set $\mathcal{B}$ of representatives of the equivalence classes of generators of power integral bases of $\mathbb{Z}[\gamma]$.

[Output] The sets $\mathcal{F}_0(\gamma)$ and $\mathcal{F}_1(\gamma)$.

(1.) [Initialize] Set $\{\beta_1, \ldots, \beta_t\} = \mathcal{B} \cup (-\mathcal{B})$, $F_0 = F_1 = T = \emptyset$ and $i = 1$.

(2.) [Compute minimal polynomial] Compute $P = \mu_{\beta_i}$.

(3.) [Element of $F_0 \cup F_1$ found?] If there exist $k \in \mathbb{Z}, \delta \in \{0, 1\}$ with $(P, k, \delta) \in T$ insert $\beta_i - k$ into F_δ and go to step 11.

(4.) [Determine upper and lower bounds] Calculate k_{β_i} and c_{β_i}.

(5.) [Insert element into F_1?] If $k_{\beta_i} - c_{\beta_i} \leq 1$ insert $\beta_i - c_{\beta_i}$ into F_1, $(P, c_{\beta_i}, 1)$ into T and go to step 11, else perform step 6 for $l = c_{\beta_i} + 1, \ldots, k_{\beta_i} - 1$, put $p_{k_{\beta_i}} = 1, k = c_{\beta_i}$ and go to step 8.

(6.) [Check CNS property] If $P(X + l) \in \mathcal{C}$ set $p_l = 1$, otherwise set $p_l = 0$.

(7.) [Check CNS basis condition] If $p_k = 0$ then go to step 9.

(8.) [Insert element into $F_0 \cup F_1$] If $p_{k+1} = \cdots = p_{k_{\beta_i}} = 1$ insert $\beta_i - k$ into F_1, $(P, k, 1)$ into T and go to step 11, else insert $\beta_i - k$ into F_0 and $(P, k, 0)$ into T.

(9.) [Next value of k] Set $k \leftarrow k + 1$.

(10.) [CNS basis check finished?] If $k \leq k_{\beta_i} - 1$ then go to step 7.

(11.) [Next generator] Set $i \leftarrow i + 1$.

(12.) [Finish?] If $i \leq t$ then go to step 2.

(13.) [Terminate] Output $\mathcal{F}_0(\gamma) = F_0$ and $\mathcal{F}_1(\gamma) = F_1$ and terminate the algorithm.

in some cases (see e.g. Theorem 2.2.1) the amount of computations can thereby be reduced. Recall that algebraic integers α, β are called equivalent if there is some $z \in \mathbb{Z}$ such that $\beta = z \pm \alpha$ (see e.g. [29]).

We verify that the algorithm delivers all CNS bases of a given order $\mathbb{Z}[\gamma]$.

2.1.11 Theorem. *Let γ be a nonzero algebraic integer and $\mathcal{B}$ a set of representatives of the equivalence classes of generators of power integral bases of $\mathbb{Z}[\gamma]$. Then algorithm computes the sets $\mathcal{F}_0(\gamma), \mathcal{F}_1(\gamma)$ with properties (i) and (ii) of Theorem 1..*

Proof It is easy to see that $\mathcal{F}_0(\gamma) \cup \mathcal{F}_1(\gamma) \subset \mathcal{C}_\gamma$ and that $\mathcal{F}_1(\gamma)$ consists of fundamental CNS bases for $\mathbb{Z}[\gamma]$. Let $\alpha \in \mathcal{C}_\gamma$, hence $\alpha = n + \beta$ with some $n \in \mathbb{Z}, \beta \in \mathcal{B} \cup (-\mathcal{B})$. Clearly, $-n \geq c_\beta$. By construction there is some integer $k \in [c_\beta, k_\beta]$ with $\beta - k \in \mathcal{F}_0(\gamma) \cup \mathcal{F}_1(\gamma)$. Let $l_1, \ldots, l_s \in [c_\beta, k_\beta]$ be exactly those indices with $p_{l_\sigma} = 0 \quad (\sigma = 1, \ldots, s)$ and $c_\beta < p_1 < \ldots < p_s < k_\beta$. If $-n \geq l_s + 1$ then $\varphi = \beta - (l_s + 1) \in \mathcal{F}_1(\gamma)$ and $\alpha = \varphi - (-n - (l_s + 1))$. Finally, let $-n < l_s + 1$, and observe that $-n \notin \{l_1, \ldots, l_s\}$. Then $-n < l_1$ or $l_\sigma < -n < l_{\sigma+1}$ for some $\sigma \in \{1, \ldots, s-1\}$ imply $\alpha \in \mathcal{F}_0(\gamma)$. □

The following example illustrates the application of algorithm . For polynomials outside the set $\mathcal{K}$ the CNS property was checked by the algorithm described in [15] (an improved version of this algorithm was implemented by T. Borbély [13]).

2.1.12 Remark. Note that if $c_\beta < k_\beta$ and $\mu_\beta(X + k) \in \mathcal{C}$ for all $k \in \{c_\beta + 1, \ldots, k_\beta - 1\}$ then $-c_\beta + \beta \in \mathcal{F}_1(\gamma)$.

2.1.13 Lemma. *Let $k \in \mathbb{Z}$.*

(i) For $f_k = f(X + k)$ with $f = X^3 - X + 3 \in \mathbb{Z}[X]$ we have

$$f_k \in \mathcal{K} \iff k \geq 3$$

and

$$f_k \in \mathcal{C} \iff k = 0 \text{ or } k \geq 2.$$

(ii) For $f_k = f(X+k)$ *with* $f = X^3 - X - 3 \in \mathbb{Z}[X]$ *we have*

$$f_k \in \mathcal{K} \iff k \geq 4$$

and

$$f_k \in \mathcal{C} \iff k \geq 3.$$

(iii) For $f_k = f(X+k)$ *with* $f = X^3 - 2X^2 - 69X - 369 \in \mathbb{Z}[X]$ *we have*

$$f_k \in \mathcal{K} \iff k \geq 13 \iff f_k \in \mathcal{C}.$$

(iv) For $f_k = f(X+k)$ *with* $f = X^3 + 2X^2 - 69X + 369 \in \mathbb{Z}[X]$ *we have*

$$f_k \in \mathcal{K} \iff k \geq 5$$

and

$$f_k \in \mathcal{C} \iff k \geq 4.$$

Proof (i) The first statement is clear because $f_k = X^3 + 3kX^2 + (3k^2 - 1)X + k^3 - k + 3$. Using this, Gilbert's theorem (see [5], Theorem 3.1) and ([5], Proposition 3.12) the second statement follows.
(ii) The first statement is clear because $f_k = X^3 + 3kX^2 + (3k^2 - 1)X + k^3 - k - 3$. Using this and Gilbert's theorem (see [5], Theorem 3.1) and checking $f_3 \in \mathcal{C}$ the second statement follows.
(iii) Clearly, $k < 13$ implies $f_k = X^3 + (3k-2)X^2 + (3k^2 - 4k - 69)X + k^3 - 2k^2 - 69k - 369 \notin \mathcal{K} \cup \mathcal{C}$.
(iv) Observing $f_k = X^3 + (3k+2)X^2 - (3k^2 + 4k - 69)X + k^3 + 2k^2 - 69k + 369$ and checking $f_4 \in \mathcal{C}$ these statements can be proved analogously. □

For a monogenic algebraic number field K we write $\mathcal{F}_\delta(K)$ instead of $\mathcal{F}_\delta(\gamma)$ where γ is some generator of a power integral basis of K $(\delta \in \{0, 1\})$.

2.1.14 Example. Let ϑ be a root of the polynomial $X^3 - X + 3 \in \mathbb{Z}[X]$. By ([29], Section 11.1) up to equivalence all generators of power integral bases of $\mathbb{Z}[\vartheta]$ are given by ϑ and $-5\vartheta + 3\vartheta^2$. By Lemma 2.1.13 we have $c_\vartheta = 0, k_\vartheta = 3$, and therefore by algorithm

$$\vartheta \in \mathcal{F}_0(\mathbb{Q}(\vartheta)), -2 + \vartheta \in \mathcal{F}_1(\mathbb{Q}(\vartheta)).$$

Analogously, we have $\mu_{-\vartheta} = X^3 - X - 3, c_{-\vartheta} = 3, k_{-\vartheta} = 4$, and then

$$-3 + \vartheta \in \mathcal{F}_1(\mathbb{Q}(\vartheta)).$$

Similarly, we have $\mu_{-5\vartheta+\vartheta^2} = X^3 - 2X^2 - 69X - 369, c_{-5\vartheta+\vartheta^2} = k_{-5\vartheta+\vartheta^2} = 13$, and

$$-13 - 5\vartheta + \vartheta^2 \in \mathcal{F}_1(\mathbb{Q}(\vartheta)),$$

and finally $\mu_{5\vartheta-\vartheta^2} = X^3 + 2X^2 - 69X + 369, c_{5\vartheta-\vartheta^2} = 4, k_{5\vartheta-\vartheta^2} = 5$, and

$$-4 + 5\vartheta - \vartheta^2 \in \mathcal{F}_1(\mathbb{Q}(\vartheta)).$$

Collecting our results we find $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \{\vartheta\}$ and

$$\mathcal{F}_1(\mathbb{Q}(\vartheta)) = \{-2 + \vartheta, -3 - \vartheta, -13 - 5\vartheta + \vartheta^2, -4 + 5\vartheta - \vartheta^2\}.$$

In some cases the determination of CNS bases is considerably easier if γ is an algebraic integer with at least one real conjugate. We then denote by $M(\gamma)$ ($m(\gamma)$) the integer part of the maximum (minimum) of the real conjugates of γ.

2.1.15 Proposition. *Let γ be a nonzero algebraic integer with at least one real conjugate and $\mathcal{B}$ a set of representatives of the equivalence classes of generators of power integral bases of $\mathbb{Z}[\gamma]$.*

(i) For $\alpha \in \mathbb{Z}[\gamma] \setminus \{0\}$ we have $c_\alpha \geq M(\alpha) + 2$ and $c_{-\alpha} \geq -m(\alpha) + 1$.

(ii) Let $\beta \in \mathcal{B}$. Then $\beta - M(\beta) - 2 \in \mathcal{F}_1(\gamma)$ if $\mu_{\beta-M(\beta)-2} \in \mathcal{K}$, and $-\beta + m(\beta) - 1 \in \mathcal{F}_1(\gamma)$ if $\mu_{-\beta+m(\beta)-1} \in \mathcal{K}$.

(iii) If $\mu_{\beta-M(\beta)-2}, \mu_{-\beta+m(\beta)-1} \in \mathcal{K}$ for all $\beta \in \mathcal{B}$ then we have $\mathcal{F}_0(\gamma) = \emptyset$ and

$$\mathcal{F}_1(\gamma) = \big\{\beta - M(\beta) - 2, -\beta + m(\beta) - 1 \,|\, \beta \in \mathcal{B}\big\}.$$

Proof (0) For every $\alpha \in \mathbb{Z}[\gamma]$ we have real embeddings $\tilde{\tau}_\alpha, \rho_\alpha$ of $\mathbb{Q}(\gamma)$ with

$$M(\alpha) \leq \tilde{\tau}_\alpha(\alpha), \quad \rho_\alpha(\alpha) < m(\alpha) + 1.$$

(i) Assume $c_\alpha = M(\alpha)+2-k$ for some $k \in \mathbb{N}\setminus\{0\}$. Then $\mu_\alpha(X+M(\alpha)+2-k) \in \mathcal{C}$, thus by ([1], Theorem 2.1)

$$\tilde{\tau}_\alpha(\alpha) - (M(\alpha)+2-k) < -1$$

which by (0) yields the contradiction

$$M(\alpha) < M(\alpha) - k + 1.$$

The other inequality is proved analogously.
(ii) It is enough to show that $(\beta - M(\beta) - 2)+1, (-\beta + m(\beta) - 1) + 1 \notin \mathcal{C}$. In view of ([1], Theorem 2.1) this is clear because by (0)

$$\tilde{\tau}_\beta(\beta - M(\beta) - 1) = \tilde{\tau}_\beta(\beta) - M(\beta) - 1 \geq M(\beta) - M(\beta) - 1 = -1,$$

$$\rho_\beta(-\beta + m(\beta)) > -m(\beta) - 1 + m(\beta) = -1.$$

(iii) Denoting by $F = \big\{\beta - M(\beta) - 2, -\beta + m(\beta) - 1 \,|\, \beta \in \mathcal{B}\big\}$ it suffices to show that

$$\mathcal{C}_\gamma \subset \big\{\varphi - n \,|\, \varphi \in F, n \in \mathbb{N}\big\}.$$

Let $\alpha \in \mathcal{C}_\gamma, \beta \in \mathcal{B}, n \in \mathbb{Z}$ with $\alpha = n \pm \beta$. In case $\alpha = n + \beta$ we have $-M(\beta) - 2 - n \in \mathbb{N}$ by (0) and

$$\alpha + (-M(\beta) - 2 - n) = \beta - M(\beta) - 2 \in F,$$

and in case $\alpha = n - \beta$ we analogously find $m(\beta) - 1 - n \in \mathbb{N}$ and

$$\alpha + (m(\beta) - 1 - n) = -\beta + m(\beta) - 1 \in F.$$

□

2.2 CNS bases in quadratic and cubic number fields

We conclude our observations by computing $\mathcal{F}_0$ and $\mathcal{F}_1$ of several quadratic, cubic and quartic number fields. For the sake of completeness we start with the formulation of some well-known results in our language.

CNS bases of quadratic number fields were studied by several authors (see [43],[44],[30],[33],[86],[8] and others).

2.2.1 Theorem. *(I. Kátai – B. Kovács, W. J. Gilbert) Let $D \neq 0, 1$ be a square-free rational integer and $\vartheta = \sqrt{D}$. Then $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ and $\mathcal{F}_1(\mathbb{Q}(\vartheta)) =$*

$$\begin{cases} \left\{-\left\lfloor \frac{1+\sqrt{D}}{2} \right\rfloor + \frac{-3+\vartheta}{2}, \left\lfloor \frac{1-\sqrt{D}}{2} \right\rfloor - \frac{3+\vartheta}{2}\right\} & , \text{ if } D > 0, D \equiv 1 \pmod 4, \\ \left\{-2 - \left\lfloor \sqrt{D} \right\rfloor + \vartheta, -2 - \left\lfloor \sqrt{D} \right\rfloor - \vartheta\right\} & , \text{ if } D > 0, D \not\equiv 1 \pmod 4, \\ \left\{\frac{-3+\vartheta}{2}, -\frac{3+\vartheta}{2}\right\} & , \text{ if } D = -3, \\ \left\{\frac{1+\vartheta}{2}, \frac{1-\vartheta}{2}\right\} & , \text{ if } D < 0, D \neq -3, D \equiv 1 \pmod 4 \\ \left\{-1+\vartheta, -1-\vartheta\right\} & , \text{ if } D = -1, \\ \left\{\vartheta, -\vartheta\right\} & , \text{ if } D < 0, D \neq -1, D \not\equiv 1 \pmod 4. \end{cases}$$

Proof A representative of the generators of power integral bases of $\mathbb{Q}(\vartheta)$ is given by $\beta = \frac{1+\vartheta}{2}$ if $D \equiv 1 \pmod 4$ ($\beta = \vartheta$ if $D \not\equiv 1 \pmod 4$). If $D > 0$ we have $m(\beta) = \left\lfloor \frac{1-\sqrt{D}}{2} \right\rfloor, M(\beta) = \left\lfloor \frac{1+\sqrt{D}}{2} \right\rfloor$ for $D \equiv 1 \pmod 4$ ($m(\beta) = \left\lfloor -\sqrt{D} \right\rfloor, M(\beta) = \left\lfloor \sqrt{D} \right\rfloor$ for $D \not\equiv 1 \pmod 4$) and our assertions follow from Proposition 2.1.15 and ([30], Theorem 1). For $D < 0$ algorithm and ([30], Theorem 1) yield the assertions. □

Using a theorem of S. Körmendi [46] S. Akiyama et al. ([5], Theorem 4.5) described all CNS in a family of pure cubic number fields.

2.2.2 Theorem. *(S. Körmendi – S. Akiyama et al.) Let $m \in \mathbb{N} \setminus \{0\}$ be not divisible by 3 and $m^3 + 1$ squarefree. For $\vartheta = \sqrt[3]{m^3+1}$ we have $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ and*

$$\mathcal{F}_1(\mathbb{Q}(\vartheta)) = \{-\vartheta, -m-2+\vartheta, -2m^2-2+m\vartheta+\vartheta^2, -m^2-2-m\vartheta-\vartheta^2\}.$$

Further, S. Akiyama et al. ([5], Theorem 4.4) determined all CNS in a family of simplest cubic number fields (for details see D. Shanks [81]). We state and slightly extend their result in our context.

2.2.3 Theorem. *(S. Akiyama et al.) Let $t \in \mathbb{Z}, t \geq -1$ and ϑ denote a root of the polynomial*

$$X^3 - tX^2 - (t+3)X - 1.$$

Then we have $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ *and*

$$\mathcal{F}_1(\mathbb{Q}(\vartheta)) = \{-3-\vartheta, -t-5-t\vartheta+\vartheta^2, -1+(t+1)\vartheta-\vartheta^2\} \cup \mathcal{G} \cup \mathcal{G}_{-1} \cup \mathcal{G}_0 \cup \mathcal{G}_2$$

where

$$\mathcal{G} = \begin{cases} \{-t-3+\vartheta, -1+t\vartheta-\vartheta^2, -t-5-(t+1)\vartheta+\vartheta^2\}, \ \textit{if } t \geq 0, \\ \emptyset \ \textit{otherwise}, \end{cases}$$

$$\mathcal{G}_{-1} = \begin{cases} \{-3+\vartheta, -2-\vartheta-\vartheta^2, -5+\vartheta^2, -19+9\vartheta+4\vartheta^2, -5-9\vartheta-4\vartheta^2, \\ -22+5\vartheta+9\vartheta^2, -2-5\vartheta-9\vartheta^2, -25-4\vartheta+5\vartheta^2, 1+4\vartheta-5\vartheta^2, \\ -7-\vartheta+\vartheta^2, -1+\vartheta-\vartheta^2, -6+2\vartheta+\vartheta^2, -2-2\vartheta-\vartheta^2, \\ -6+\vartheta+2\vartheta^2, -2-\vartheta-\vartheta^2\}, \ \textit{if } t = -1, \\ \emptyset \ \textit{otherwise}, \end{cases}$$

$$\mathcal{G}_0 = \begin{cases} \{-9+2\vartheta+\vartheta^2, -2-2\vartheta-\vartheta^2, -11-3\vartheta+2\vartheta^2, -1+3\vartheta-2\vartheta^2, \\ \quad -10-\vartheta+3\vartheta^2, -1+\vartheta-3\vartheta^2\}, \ \textit{if } t = 0, \\ \emptyset \ \textit{otherwise}, \end{cases}$$

$$\mathcal{G}_2 = \begin{cases} \{-37+3\vartheta+2\vartheta^2, -2-3\vartheta-2\vartheta^2, -42-20\vartheta+9\vartheta^2, 3+20\vartheta-9\vartheta^2, \\ \quad -43-23\vartheta+7\vartheta^2, -4+23\vartheta-7\vartheta^2\}, \ \textit{if } t = 2, \\ \emptyset \ \textit{otherwise}. \end{cases}$$

Proof We proceed similarly as in Example 2.1.14, but leave the verifications of computational details to the reader. By [29] up to equivalence all generators of power integral bases of $\mathbb{Z}[\vartheta]$ are the following:

- for arbitrary t: $\vartheta, -t\vartheta+\vartheta^2, (t+1)\vartheta-\vartheta^2$;
- for $t=-1$ additionally: $9\vartheta+4\vartheta^2, 5\vartheta+9\vartheta^2, -4\vartheta+5\vartheta^2, -\vartheta+\vartheta^2, 2\vartheta+\vartheta^2, \vartheta+2\vartheta^2$;
- for $t=0$ additionally: $2\vartheta+\vartheta^2, -3\vartheta+2\vartheta^2, -\vartheta+3\vartheta^2$;
- for $t=2$ additionally: $3\vartheta+2\vartheta^2, -20\vartheta+9\vartheta^2, -23\vartheta+7\vartheta^2$.

The proof is now accomplished by Proposition 2.1.15 and Table 1 below where we use the following notation: β is a generator of a power integral basis of $\mathbb{Q}(\vartheta)$. The minimal polynomial $\mu_\beta = X^3 + a_1X^2 + a_2X + a_3$ of β is given by (a_1, a_2, a_3). Lower bounds for the constants c_β, k_β are given by Proposition 2.1.15. For their determination ([5], Theorem 3.1) and ([16], Theorem 5.1) are used. Observe that in all cases considered here Remark 2.1.12 applies if $c_\beta \leq k_\beta - 2$ or $c_{-\beta} \leq k_{-\beta} - 2$. □

β	t	μ_β	$m(\beta)$	$M(\beta)$	c_β	k_β	$c_{-\beta}$	$k_{-\beta}$
ϑ	≥ 5	$(-t, -t-3, -1)$	-2	$t{+}1$	$t{+}3$	$t{+}3$	3	3
ϑ	$0, \ldots, 4$	$(-t, -t-3, -1)$	-2	$t{+}1$	$t{+}3$	$t{+}3$	3	4
ϑ	-1	$(1, -2, -1)$	-2	1	3	4	3	4
$-t\vartheta + \vartheta^2$	≥ 5	$(-2t{-}6, t^2{+}7t{+}9, -t^2 - 3t - 1)$	0	$t{+}3$	$t{+}5$	$t{+}5$	1	1
$-t\vartheta + \vartheta^2$	2, 3, 4	$(-2t{-}6, t^2{+}7t{+}9, -t^2 - 3t - 1)$	0	$t{+}3$	$t{+}5$	$t{+}6$	1	1
$-\vartheta + \vartheta^2$	1	$(-8, 17, -5)$	0	4	6	7	1	2
ϑ^2	0	$(-6, 9, -1)$	0	3	5	6	1	2
$\vartheta + \vartheta^2$	-1	$(-4, 3, 1)$	-1	2	4	5	2	3
$(t+1)\vartheta - \vartheta^2$	≥ 3	$(t{+}6, 3t{+}9, 2t{+}3)$	$-t{-}4$	-1	1	2	$t{+}5$	$t{+}5$
$(t+1)\vartheta - \vartheta^2$	0, 1, 2	$(t{+}6, 3t{+}9, 2t{+}3)$	$-t{-}4$	-1	1	2	$t{+}5$	$t{+}6$
$-\vartheta^2$	-1	$(5, 6, 1)$	-4	-1	1	3	5	6
$3\vartheta + 2\vartheta^2$	2	$(-34, -39, -11)$	-1	35	37	37	2	3
$-20\vartheta + 9\vartheta^2$	2	$(-86, 2041, -8029)$	4	40	42	43	-3	-3
$-23\vartheta + 7\vartheta^2$	2	$(-52, 477, -1217)$	5	41	43	43	-4	-3
$9\vartheta + 4\vartheta^2$	-1	$(-11, -102, -181)$	-4	17	19	19	5	6
$5\vartheta + 9\vartheta^2$	-1	$(-40, 391, 181)$	-1	20	22	23	2	2
$-4\vartheta + 5\vartheta^2$	-1	$(-29, 138, -181)$	2	23	25	25	-1	0
$-\vartheta + \vartheta^2$	-1	$(-6, 5, -1)$	0	5	7	7	1	2
$2\vartheta + \vartheta^2$	0	$(-6, -9, -3)$	-1	7	9	9	2	3
$2\vartheta + \vartheta^2$	-1	$(-3, -4, -1)$	-1	4	6	6	2	3
$-3\vartheta + 2\vartheta^2$	0	$(-12, 27, -17)$	1	9	11	11	0	1
$-\vartheta + 3\vartheta^2$	0	$(-18, 87, -53)$	0	8	10	11	1	1
$\vartheta + 2\vartheta^2$	-1	$(-9, 20, 1)$	-1	4	6	7	2	2

TABLE 1

2.3 CNS bases in quartic cyclotomic fields

In this section we treat the cyclotomic fields of degree 4.

2.3.1 Theorem. *Let ζ be a primitive eighth root of unity. Then we have $\mathcal{F}_0(\mathbb{Q}(\zeta)) = \emptyset$ and*

$$\mathcal{F}_1(\mathbb{Q}(\zeta)) = \{-3 \pm \zeta^k \mid k = 1, 3, 5, 7\}.$$

Proof By R. Robertson [73] up to equivalence all generators of power integral bases of $\mathbb{Q}(\zeta)$ are given by $\zeta^k, k \in \mathbb{Z}, k$ odd. Observing $\mu_\zeta = X^4 + 1$ one immediately finds $k_\zeta = 4$. The algorithm described in [15] and ([8], Theorem 5.4) yield $c_\zeta = 3$, and a straightforward application of algorithm concludes the proof. □

2.3.2 Theorem. *Let ζ be a primitive twelfth root of unity. Then we have $\mathcal{F}_0(\mathbb{Q}(\zeta)) = \emptyset$ and*

$$\mathcal{F}_1(\mathbb{Q}(\zeta)) = \{-3+\zeta, -3-\zeta, -3+\zeta^{-1}, -3-\zeta^{-1}, -1-\zeta^2+\zeta^{-1}, -2+\zeta^2-\zeta^{-1}\}.$$

Proof The proof works analogously as that of Theorem 2.. □

2.3.3 Theorem. *Let ζ be a primitive fifth root of unity. Then we have $\mathcal{F}_0(\mathbb{Q}(\zeta)) = \emptyset$ and*

$$\mathcal{F}_1(\mathbb{Q}(\zeta)) = \{-2 + \zeta, -3 - \zeta, -2 + \zeta + \zeta^3, -3 - \zeta - \zeta^3\}.$$

Proof By [72] up to equivalence all generators of power integral bases of $\mathbb{Z}[\zeta]$ are ζ and $\frac{1}{1+\zeta}$. One immediately checks that

$$f_k(X) = \mu_\zeta(X + k) \in \mathcal{K} \iff k \geq 4,$$

hence $k_\zeta = 4$. By ([8], Theorem 5.4) one finds $k \geq -5$ for $f_k \in \mathcal{C}$. Trivially, $f_0, f_{-1} \notin \mathcal{C}$, and an application of the algorithm described in [15] yields $f_k \notin \mathcal{C}$ for $k = -5, -4, -3, -2, 1$, but $f_2, f_3 \in \mathcal{C}$. Thus we have shown that

$$f_k \in \mathcal{C} \iff k \geq 2,$$

hence $c_\zeta = 2$ and $f_k \in \mathcal{C}$ for all $k \in \{c_\zeta, \ldots, k_\zeta\}$.

β	μ_β	c_β	k_β	$c_{-\beta}$	$k_{-\beta}$
ζ	$(1,1,1,1)$	2	4	3	5
$-\zeta-\zeta^3$	$(-2,4,-3,1)$	3	5	2	4

TABLE 2

Therefore by algorithm we find $-2+\zeta \in \mathcal{F}_1(\mathbb{Q}(\zeta))$. Similarly, the other cases are dealt with. The main data are listed in Table 2 below where we use the following notation: β is a generator of a power integral basis of $\mathbb{Q}(\zeta)$, the minimal polynomial $\mu_\beta = X^4 + a_1X^3 + a_2X^2 + a_3X + a_4$ of β is given by (a_1, a_2, a_3, a_4). □

2.4 CNS bases in quartic number fields

For the convenience of the reader we rephrase a result of A. Pethő ([67], Theorem 15) in our settings.

2.4.1 Theorem. *(A. Pethő) Let $f \in \mathbb{N}, f \geq 3, f$ odd, $m = f^2 + 2$ and $n = f^2 - 2$. Then we have $\mathcal{F}_0(\mathbb{Q}(\sqrt{m}, \sqrt{n})) = \emptyset$ and*

$$\mathcal{F}_1(\mathbb{Q}(\sqrt{m}, \sqrt{n})) = \{-f-1+\vartheta_1, -f-1-\vartheta_1, -1-\frac{3f^3+f}{2}+\vartheta_2, -2-\frac{f^3-f}{2}-\vartheta_2\}$$

where

$$\vartheta_1 = \frac{\sqrt{m}+\sqrt{n}}{2}, \quad \vartheta_2 = f\frac{1+\sqrt{mn}}{2} + \sqrt{n} + (f^2-1)\frac{\sqrt{m}+\sqrt{n}}{2}.$$

For $t \in \mathbb{Z} \setminus \{0, \pm 3\}$ let

$$P_t(X) = X^4 - tX^3 - 6X^2 + tX + 1.$$

Let $\vartheta = \vartheta_t$ be a root of $P_t(X)$, then the infinite parametric family of number fields $K_t = K = Q(\vartheta_t)$ is called *simplest quartic fields*. P. Olajos [57] proved that K_t admits a power integral bases if and only if $t = 2$ and $t = 4$, moreover he found all generators of power integral bases in these fields. Using his result we are able to compute all CNS bases in such fields.

2.4.2 Theorem. *We have* $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ *and* $\mathcal{F}_1(\mathbb{Q}(\vartheta_2)) = \mathcal{G}_2$ *and* $\mathcal{F}_1(\mathbb{Q}(\vartheta_4)) = \mathcal{G}_4$ *where*

$$\begin{aligned}
\mathcal{G}_2 &= \{-\frac{1}{2}\vartheta^3+\vartheta^2+\frac{7}{2}\vartheta-4, \frac{1}{2}\vartheta^3-\vartheta^2-\frac{7}{2}\vartheta-2, 2\vartheta^3-\frac{9}{2}\vartheta^2-11\vartheta-\frac{9}{2},\\
&\quad -2\vartheta^3+\frac{9}{2}\vartheta^2+11\vartheta-\frac{19}{2}, \frac{1}{2}\vartheta^3-2\vartheta-\frac{13}{2}, -\frac{1}{2}\vartheta^3+2\vartheta-\frac{5}{2},\\
&\quad \frac{1}{2}\vartheta^2+\vartheta-\frac{23}{2}, -\frac{1}{2}\vartheta^2-\vartheta-\frac{5}{2}, \vartheta^3-\frac{3}{2}\vartheta^2-7\vartheta-\frac{9}{2},\\
&\quad -\vartheta^3+\frac{3}{2}\vartheta^2+7\vartheta-\frac{11}{2}, \frac{3}{2}\vartheta^3-2\vartheta^2-\frac{21}{2}\vartheta-6,\\
&\quad -\frac{3}{2}\vartheta^3+2\vartheta^2+\frac{21}{2}\vartheta-8, \frac{1}{2}\vartheta^3-2\vartheta^2+\frac{1}{2}\vartheta-1, -\frac{1}{2}\vartheta^3+2\vartheta^2-\frac{1}{2}\vartheta-11,\\
&\quad -\vartheta^3+\frac{5}{2}\vartheta^2+5\vartheta-\frac{13}{2}\vartheta^3-\frac{5}{2}\vartheta^2-5\vartheta-\frac{5}{2}, \frac{1}{2}\vartheta^2-\vartheta-\frac{9}{2},\\
&\quad -\frac{1}{2}\vartheta^2+\vartheta-\frac{3}{2}, \frac{1}{2}\vartheta^2-\frac{15}{2}, -\frac{1}{2}\vartheta^2-\frac{3}{2}\}\\
\mathcal{G}_4 &= \Big\{-\frac{1}{4}\vartheta^3+\frac{3}{4}\vartheta^2+\frac{11}{4}\vartheta-\frac{13}{4}, \frac{1}{4}\vartheta^3-\frac{3}{4}\vartheta^2-\frac{11}{4}\vartheta-\frac{11}{4},\\
&\quad \frac{1}{4}\vartheta^3-\frac{3}{4}\vartheta^2-\frac{7}{4}\vartheta-\frac{23}{4}, -\frac{1}{4}\vartheta^3+\frac{3}{4}\vartheta^2+\frac{7}{4}\vartheta-\frac{13}{4},\\
&\quad -\frac{3}{4}\vartheta^3+\frac{13}{4}\vartheta^2+\frac{13}{4}\vartheta-\frac{27}{4}, \frac{3}{4}\vartheta^3-\frac{13}{4}\vartheta^2-\frac{13}{4}\vartheta-\frac{9}{4},\\
&\quad \frac{3}{4}\vartheta^3-\frac{11}{4}\vartheta^2-\frac{21}{4}\vartheta-\frac{11}{4}, -\frac{3}{4}\vartheta^3+\frac{11}{4}\vartheta^2+\frac{21}{4}\vartheta-\frac{25}{4},\\
&\quad -\frac{1}{4}\vartheta^3+\frac{5}{4}\vartheta^2-\frac{1}{4}\vartheta-\frac{23}{4}, \frac{1}{4}\vartheta^3-\frac{5}{4}\vartheta^2+\frac{1}{4}\vartheta-\frac{13}{4},\\
&\quad -\frac{1}{4}\vartheta^3+\frac{5}{4}\vartheta^2+\frac{3}{4}\vartheta-\frac{19}{4}, \frac{1}{4}\vartheta^3-\frac{5}{4}\vartheta^2-\frac{3}{4}\vartheta-\frac{5}{4}\}.
\end{aligned}$$

Proof Let γ be a generator of power integral basis in $\mathbb{Z}_K$. P. Olajos [57] showed that only the following cases can occur:

- $t=2, \gamma = x\cdot\vartheta + y\cdot\frac{1+\vartheta^2}{2} + z\cdot\frac{\vartheta+\vartheta^3}{2}$ where $(x,y,z) = (4,2,-1), (-13,-9,4), (-2,1,0), (1,1,0), (-8,-3,2), (-12,-4,3), (0,-4,1), (6,5,-2), (-1,1,0), (0,1,0)$.

- $t = 4, \gamma = x \cdot \vartheta + y \cdot \frac{1+\vartheta^2}{2} + z \cdot \frac{1+\vartheta+\vartheta^2+\vartheta^3}{4}$ where
  $(x, y, z) = (3, 2, -1), (-2, -2, 1), (4, 8, -3), (-6, -7, 3), (0, 3, -1), (1, 3, -1).$

From here on we proceed as in the proof of Theorem 4.. The details of the computation are given in Table 3 below where we use the following notation: (x, y, z) denote the coordinates of γ as in the table above, the minimal polynomial $\mu_\gamma = X^4 + a_1X^3 + a_2X^2 + a_3X + a_4$ of γ is given by (a_1, a_2, a_3, a_4).

(x, y, z)	γ	μ_γ	c_γ	k_γ	$c_{-\gamma}$	$k_{-\gamma}$
$(4, 2, -1)$	$-\frac{1}{2}\vartheta^3 + \vartheta^2 + \frac{7}{2}\vartheta + 1$	$(-8, 19, -12, 1)$	5	7	1	3
$(-13, -9, 4)$	$2\vartheta^3 - \frac{9}{2}\vartheta^2 - 11\vartheta - \frac{9}{2}$	$(36, 451, 2176, 2641)$	0	0	14	15
$(-2, 1, 0)$	$\frac{1}{2}\vartheta^3 - 2\vartheta + \frac{1}{2}$	$(-6, 1, 4, 1)$	7	8	2	4
$(1, 1, 0)$	$\frac{1}{2}\vartheta^2 + \vartheta + \frac{1}{2}$	$(-12, 19, -8, 1)$	12	12	2,	3
$(-8, -3, 2)$	$\vartheta^3 - \frac{3}{2}\vartheta^2 - 7\vartheta - \frac{3}{2}$	$(6, 1, -4, 1)$	2	4	7	8
$(-12, -4, 3)$	$\frac{3}{2}\vartheta^3 - 2\vartheta^2 - \frac{21}{2}\vartheta - 2$	$(4, -29, 44, -19)$	4	5	10	10
$(0, -4, 1)$	$\frac{1}{2}\vartheta^3 - 2\vartheta^2 + \frac{1}{2}\vartheta - 2$	$(20, 115, 260, 205)$	0	1	14	14
$(6, 5, -2)$	$-\vartheta^3 + \frac{5}{2}\vartheta^2 + 5\vartheta + \frac{5}{2}$	$(-22, 169, -508, 421)$	9	11	0	1
$(-1, 1, 0)$	$\frac{1}{2}\vartheta^2 - \vartheta + \frac{1}{2}$	$(-8, 19, -12, 1)$	5	7	1	3
$(0, 1, 0)$	$\frac{1}{2}\vartheta^2 + \frac{1}{2}$	$(-10, 25, -20, 5)$	8	9	1	3
$(3, 2, -1)$	$-\frac{1}{4}\vartheta^3 + \frac{3}{4}\vartheta^2 + \frac{11}{4}\vartheta + \frac{3}{4}$	$(-4, 2, 4, -1)$	4	6	2	4
$(-2, -2, 1)$	$\frac{1}{4}\vartheta^3 - \frac{3}{4}\vartheta^2 - \frac{7}{4}\vartheta - \frac{3}{4}$	$(0, -8, -8, -2)$	5	6	4	5
$(4, 8, -3)$	$-\frac{3}{4}\vartheta^3 + \frac{13}{4}\vartheta^2 + \frac{13}{4}\vartheta + \frac{13}{4}$	$(-24, 208, -760, 958)$	10	11	-1	0
$(-6, -7, 3)$	$\frac{3}{4}\vartheta^3 - \frac{11}{4}\vartheta^2 - \frac{21}{4}\vartheta - \frac{11}{4}$	$(16, 88, 200, 158)$	0	1	9	10
$(0, 3, -1)$	$-\frac{1}{4}\vartheta^3 + \frac{5}{4}\vartheta^2 - \frac{1}{4}\vartheta + \frac{5}{4}$	$(-8, 16, -8, -2)$	7	8	2	3
$(1, 3, -1)$	$-\frac{1}{4}\vartheta^3 + \frac{5}{4}\vartheta^2 + \frac{3}{4}\vartheta + \frac{5}{4}$	$(-12, 50, -84, 47)$	6	8	0	2

TABLE 3

□

Power integral bases in the polynomial order $\mathbb{Z}[\alpha]$ of K_t were described by G. Lettl and A. Pethő [53].

2.4.3 Theorem. *Let $t \in \mathbb{N} \setminus \{0, 3\}$ and ϑ denote a root of the polynomial*

$$X^4 - tX^3 - 6X^2 + tX + 1.$$

Then we have $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ *and* $\mathcal{F}_1(\mathbb{Q}(\vartheta)) = \mathcal{G} \cup \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_4$ *where*

$$\mathcal{G} = \begin{cases} \{-3-\vartheta, -t-2+\vartheta, -2-6\vartheta-t\vartheta^2+\vartheta^3, \\ -t-3+6\vartheta+t\vartheta^2-\vartheta^3\}, \ \textit{if } t \geq 5, \\ \emptyset \ \textit{otherwise}, \end{cases}$$

$$\mathcal{G}_1 = \begin{cases} \{-4+\vartheta, -4-\vartheta, -5+6\vartheta+\vartheta^2-\vartheta^3, -3-6\vartheta-\vartheta^2+\vartheta^3, \\ -23+3\vartheta^2-\vartheta^3, \quad -1-3\vartheta^2+\vartheta^3, -14+25\vartheta+2\vartheta^2-4\vartheta^3, \\ -10-25\vartheta-2\vartheta^2+4\vartheta^3\}, \ \textit{if } t = 1, \\ \emptyset \ \textit{otherwise}, \end{cases}$$

$$\mathcal{G}_2 = \begin{cases} \{-5+\vartheta, -3-\vartheta, -5+6\vartheta+2\vartheta^2-\vartheta^3, -3-6\vartheta-2\vartheta^2+\vartheta^3\}, \ \textit{if } t = 2, \\ \emptyset \ \textit{otherwise}, \end{cases}$$

$$\mathcal{G}_4 = \begin{cases} \{-6+\vartheta, -3-\vartheta, 1+9\vartheta-22\vartheta^2+4\vartheta^3, -78-9\vartheta+22\vartheta^2-4\vartheta^3, \\ \quad -7+6\vartheta+4\vartheta^2-\vartheta^3, -3-6\vartheta-4\vartheta^2+\vartheta^3, -62+74\vartheta+30\vartheta^2-9\vartheta^3, \\ \quad -15-74\vartheta-30\vartheta^2+9\vartheta^3\}, \ \textit{if } t = 4, \\ \emptyset \ \textit{otherwise}. \end{cases}$$

Before embarking on the proof of Theorem 4. we need some preparation. For checking the CNS property of some polynomials we exploit a technical lemma which we state in a more general form without any extra amount of effort. For the notation the reader is referred to [2].

2.4.4 Lemma. *The vector* $\mathbf{r} = (r_1, \ldots, r_4) \in \mathbb{R}^4$ *with the properties*

(i) $r_2 \geq 2r_1 > 0$

(ii) $r_4 \geq 1 + r_1$

(iii) $r_1 + 2r_3 - r_4 \leq 0$

(iv) $2r_2 - r_3 + 2r_4 < 2$

belongs to $\mathcal{D}_4^0$.

Proof Let

$$E = \{(e_1, \ldots, e_4) \in \mathbb{Z}^4 \,|\, |e_i| \le 2 \quad (i = 1, \ldots, 4), \quad (e_3, e_4) \neq (0, \pm 2),$$

$$e_i e_{i+1} \le 0 \quad (i = 1, 2, 3), \quad |e_i| = 2 \implies e_{i+1} \neq 0 \quad (i = 1, 2, 3)\}$$

and $\tilde{\tau}_{\mathbf{r}}(a) = (a_2, a_3, a_4, -\lfloor r_1 a_1 + \cdots + r_4 a_4 \rfloor)$ be a mapping on $\mathbb{Z}^4$. Clearly, property (i) of ([2], Theorem 5.1) is satisfied. We show (ii) and (iii) of ([2], Theorem 5.1) in several steps thereby using the notation of ([66], Lemma 1): $a \xrightarrow{(S)}$ indicates that $\tilde{\tau}_{\mathbf{r}}(a)$ falls into step(s) S considered before.

(1) $e_1 \ge 0, \tilde{\tau}_{\mathbf{r}}(e_1, 0, 0, 0) = 0$

(2) $e_1 \le 0, (e_1, 1, 0, 0) \xrightarrow{(1)}$

(3) $(e_1, -1, 1, 0) \xrightarrow{(2)}$

(4) $e_2 \in \{0, 1\}, (e_1, e_2, -1, 1) \xrightarrow{(3)}$

(5) $(e_1, -1, 1, -1) \xrightarrow{(4)}$

(6) $(e_1, 2, -1, 1) \xrightarrow{(3,5)}$

(7) $(e_1, 0, 1, -1) \xrightarrow{(4)}$

(8) $e_2 \in \{0, 1\}, (e_1, e_2, 0, 1) \xrightarrow{(7)}$

(9) $(e_1, e_2, 0, 0) \xrightarrow{(1,8)}$

(10) $(e_1, 0, 1, 0) \xrightarrow{(9)}$

(11) $(e_1, -1, 0, 1) \xrightarrow{(7,10)}$

(12) $(e_1, 2, -1, 0) \xrightarrow{(11)}$

(13) $(e_1, -1, 2, -1) \xrightarrow{(6,12)}$

(14) $(e_1, 1, -1, 2) \xrightarrow{(13)}$

(15) $(e_1, e_2, 1, -1) \xrightarrow{(4,5,7,14)}$

(16) $e_1 \le -1, (e_1, e_2, 2, -1) \xrightarrow{(6,12)}$

(17) $(e_1, 0, -1, 2) \xrightarrow{(16)}$

(18) $(e_1, 1, 0, -1) \xrightarrow{(4,17)}$

(19) $(e_1, e_2, 1, 0) \xrightarrow{(9)}$

(20) $(e_1, e_2, -1, 0) \xrightarrow{(11)}$
(21) $(e_1, e_2, e_3, 0) \xrightarrow{(9,19,20)}$
(22) $e_4 \geq 1, (e_1, e_2, e_3, e_4) \xrightarrow{(13,15,21)}$
(23) $(e_1, 2, -1, 1) \xrightarrow{(21)}$
(24) $(e_1, e_2, e_3, -1) \xrightarrow{(4,6,17)}$
(25) $(e_1, e_2, e_3, e_4) \xrightarrow{(21,22,24)}$
This concludes the proof. □

We shall make use of the following consequence of this lemma.

2.4.5 Corollary. *The polynomial* $X^4 + p_3X^3 + p_2X^2 + p_1X + p_0 \in \mathbb{Z}[X]$ *with the properties*

(i) $p_0 \geq 2$

(ii) $p_1 \geq p_0 + 1$

(iii) $p_3 \geq 2$

(iv) $p_1 \geq 2p_3 + 1$

(v) $2p_1 - p_2 + 2p_3 \leq 2p_0 - 1$

is a CNS polynomial.

Proof This is clear by Lemma 2.4.4 and ([2], Theorem 3.1). □
We are now in a position to verify Theorem 4..

Proof of Theorem 4.. By [29] up to equivalence all generators of power integral bases of $\mathbb{Z}[\vartheta]$ are the following:

- for $t \in \mathbb{N} \setminus \{0, 3\}$: $\vartheta, 6\vartheta + t\vartheta^2 - \vartheta^3$,
- for $t = 1$ additionally: $3\vartheta^2 - \vartheta^3, 25\vartheta + 2\vartheta^2 - 4\vartheta^3$,
- for $t = 4$ additionally: $9\vartheta - 22\vartheta^2 + 4\vartheta^3, -74\vartheta - 30\vartheta^2 + 9\vartheta^3$.

We proceed analogously as in the proof of Theorem 2.2.3 by using Proposition 2.1.15 and Table 4 below with the following notation: β is a generator of a power integral basis of $\mathbb{Q}(\vartheta)$. The minimal polynomial $\mu_\beta = X^4 + a_1X^3 +$

$a_2X^2 + a_3X + a_4$ of β is listed in the form (a_1, a_2, a_3, a_4). Lower bounds for the constants c_β, k_β are given by Proposition 2.1.15. For their determination ([5], Theorem 3.1) and Corollary 2.4.5 are used in a straightforward way. Similarly as in the proof of Theorem 2.2.3 Remark 2.1.12 is used. □

β	t	μ_β	$m(\beta)$	$M(\beta)$	c_β	k_β	$c_{-\beta}$	$k_{-\beta}$
ϑ	$\neq 1, 2$	$(-t, -6, t, 1)$	-2	t	t+2	t+2	3	4
ϑ	1	$(-1, -6, 1, 1)$	-3	2	4	6	4	5
ϑ	2	$(-2, -6, 2, 1)$	-2	3	5	6	3	5
$6\vartheta + t\vartheta^2 - \vartheta^3$	$\neq$ 1, 2, 4	$(-3t, 3t^2 - 6, -t^3 + 11t, -5t^2 + 1)$	-1	t+1	t+3	t+4	2	2
$6\vartheta+\vartheta^2-\vartheta^3$	1	$(-1, -6, 1, 1)$	-3	2	4	6	4	5
$6\vartheta + 2\vartheta^2 - \vartheta^3$	2	$(-6, -6, 14, -19)$	-2	3	5	7	3	4
$6\vartheta + 4\vartheta^2 - \vartheta^3$	4	$(-12, 42, -20, -79)$	-2	5	7	8	3	3
$3\vartheta^2 - \vartheta^3$	1	$(-23, 39, -22, 4)$	0	21	23	23	1	3
$25\vartheta+2\vartheta^2-4\vartheta^3$	1	$(13, -96, -1993, -7241)$	-9	12	14	14	10	12
$9\vartheta-22\vartheta^2+4\vartheta^3$	4	$(84, 618, 1580, 1361)$	-77	-3	-1	1	78	78
$-74\vartheta - 30\vartheta^2 + 9\vartheta^3$	4	$(20, -1878, 29932, -144239)$	-61	13	15	17	62	62

TABLE 4

Finally we consider another family of orders in a parametrized family of quartic number fields, where all power integral bases are known. Let $t \in \mathbb{Z}$, $t \geq 0$, and $P(X) = X^4 - tX^3 - X^2 + tX + 1$. Denote by α one of the zeros of $P(X)$. In the following we deal with the order $\mathcal{O} = Z[\alpha]$ of $Q(\alpha)$.

M. Mignotte, A. Pethő and R. Roth [55] gave the following result:

2.4.6 Theorem. *(M. Mignotte, A. Pethő, R. Roth) Let $t \geq 4$. Then every element $\gamma \in \mathcal{O}$ such that $Z[\gamma] = \mathcal{O}$ is equivalent to some element $\gamma = x\alpha + y\alpha^2 + z\alpha^3$ with*

$$\begin{aligned}(x, y, z) \quad \in \quad & \{(1, 0, 0), (1, t, -1), (t, t-1, -1), (t, -t-1, 1), (1, 0, -1), \\ & (1, -t(t^2+1), t^2)\}\end{aligned}$$

except when $t = 4$, in which case additionally $(x, y, z) \in \{(209, 140, -49), (209, -312, 64)\}$. [1]

2.4.7 Theorem. *Let $t \geq 4$. We have $\mathcal{F}_0(\mathbb{Q}(\alpha)) = \emptyset$ and $\mathcal{F}_1(\mathbb{Q}(\alpha)) = \mathcal{G}_4 \cup \mathcal{G}_t$ where*

$$\begin{aligned}\mathcal{G}_4 &= \left\{209\alpha + 140\alpha^2 - 49\alpha^3 + 350, 209\alpha - 312\alpha^2 + 64\alpha^3 - 71\right\}\\ \mathcal{G}_t &= \left\{\alpha + t + 1, \alpha + t\alpha^2 - \alpha^3 + t + 2, t\alpha + (t-1)\alpha^2 - \alpha^3 + 8,\right.\\ &\quad \left. t\alpha - (t+1)\alpha^2 + \alpha^3 + 2, \alpha - \alpha^3 + 2, \alpha - t(t^2+1)\alpha^2 + t^2\alpha^3 - t + 1\right\}.\end{aligned}$$

To prove this Theorem we need the some Lemmata.

2.4.8 Lemma. *If $p_0 = p_1 - p_2 + p_3$ and $p_3 < p_0 < p_2 < p_1$ and $p_0 \leq p_2 - p_3 < 2p_0$ and $p_2 - 2p_3 + 2 < p_0$, then $X^4 + p_3X^3 + p_2X^2 + p_1X + p_0$ is not a CNS polynomial.*

Proof Considering $(1, 0, -1, 2)$ and applying mapping $\tilde{\tau}$ we get $(0, -1, 2, -2)$, since $-\lfloor 2 + \frac{p_2 - 2p_3 + 1}{p_0} \rfloor = -2$. Calculating in a similar way we get the following sequence:

$$\begin{aligned}&(1,0,-1,2),(0,-1,2,-2),(-1,2,-2,2),(2,-2,2,-1),(-2,2,-1,0),\\ &(2,-1,0,1),(-1,0,1,-1),(0,1,-1,1),(1,-1,1,-1),\\ &(-1,1,-1,1),(1,-1,1,0),(-1,1,0,-1),(1,0,-1,2).\end{aligned}$$

This sequence contains a cycle starting with $(1, 0, -1, 2)$, hence polynomials with the properties above are not CNS. □

2.4.9 Lemma. *The $P(X) = X^4 + (8+t)X^3 + (23+6t)X^2 + (28+11t)X + 13 + 6t$ is a CNS polynomial for every $t \geq 4$.*

The proof of this lemma is quite complicated, therefore it can be found in Appendix.

Proof of Theorem 5.. We follow the same line as in the proof of Theorem 4.. First we compute the data necessary to apply algorithm . For the zeroes

[1] In Theorem 4 of [55] the last vector reads $(209, -352, 64)$, but its correct value is $(209, -312, 64)$.

of the polynomial $P(X)$ we use the following estimates:

$$\alpha_1 = t - 1/t^3 - 1/t^5 - 4/t^7 - 9/t^9,\ \alpha_2 = -1/t - 1/t^5 - 1/t^7 - 5/t^9,$$
$$\alpha_3 = 1 + 1/2t + 1/8t^2 + 1/2t^3,\ \alpha_4 = -1 + 1/2t - 1/8t^2.$$

In a straightforward way we obtain $M(\gamma)$ for any possible value of γ. Knowing $M(\gamma)$ it is easy to establish k_γ. Because of the special form of $P(X)$ we do not need $k_{-\gamma}$. Indeed denote by σ the automorphism of $\mathbb{Q}(\alpha)$, which maps α to $-\frac{1}{\alpha}$. Then an easy computation shows that

$$\begin{aligned} \sigma(-\alpha) &= \alpha + t\alpha^2 - \alpha^3 - t \\ \sigma(-(t\alpha + (t-1)\alpha^2 - \alpha^3)) &= t\alpha - (t-1)\alpha^2 + \alpha^3 + 1 \\ \sigma(-(\alpha - \alpha^3)) &= \alpha - t(t^2+1)\alpha^2 + t^2\alpha^3 + t^3 \end{aligned}$$

and if $t = 4$ then

$$\sigma(-(209\alpha + 140\alpha^2 - 49\alpha^3)) = 209\alpha - 312\alpha^2 + 64\alpha^3 + 116.$$

The details of the computation are given in Table 5 below where we use the following notation: (x, y, z) denote the coordinates of $\gamma = x\alpha + y\alpha^2 + z\alpha^3$ as in Theorem 2.4.6, the minimal polynomial $\mu_\gamma = X^4 + a_1X^3 + a_2X^2 + a_3X + a_4$ of γ is given by (a_1, a_2, a_3, a_4). We gave c_γ as well, although its computation is detailed after the table.

γ	μ_γ	$m(\gamma)$	$M(\gamma)$	c_γ	k_γ
α	$(-t, -1, t, 1)$	-1	$t-1$	$t+1$	$t+3$, if $t=4$ $t+2$, if $t>4$
$\alpha + t\alpha^2 - \alpha^3$	$(-3t, 3t^2-1, t-t^3, 1)$	0	t	$t+2$	$t+4$
$t\alpha + (t-1)\alpha^2 - \alpha^3$	$(2-2t, -3t+5, -t+4, 1)$	-1	6 $2t-1$	8 $2t+1$	8, if $t=4$ $2t+1$, if $t>4$
$t\alpha - (t+1)\alpha^2 + \alpha^3$	$(2t+2, 3t+5, t+4, 1)$	$-2t-1$	-2	2	3
$\alpha - \alpha^3$	$(t^3-t, 3t^2-1, 3t, 1)$	$-t^3+t$	-1	2	3
$\alpha - t(t^2+1)\alpha^2 + t^2\alpha^3$	$(3t^3+t,$ $3t^6+3t^4+3t^2-1,$ $t^9+3t^7+6t^5-2t^3-3t,$ $t^{10}+3t^8-t^6-3t^4+1)$	$-t^3-1$	$-t-1$	$-t+1$	$-t+1$
$209\alpha + 140\alpha^2 - 49\alpha^3$	(-4,2,4,-1)	-43	348	350	350
$209\alpha - 312\alpha^2 + 64\alpha^3$	(0,-8,-8,-2)	-465	-74	-71	-70

TABLE 5

As all zeroes of $P(X)$ are real, by Proposition 2.1.15 it is enough to test the polynomials $\mu_\gamma(X+n)$ for $M(\gamma)+2 \le n < k_\gamma$.

Case(1) $\gamma = \alpha$. Then

$$\mu_\gamma(X+t+1) = X^4 + (4+3t)X^3 + (5+9t+3t^2)X^2 + \\ +(2+8t+6t^2+t^3)X + 1 + 2t + 3t^2 + t^3,$$

which belongs to $\mathcal{C}$. To show this put

$$E = \{e = (-a+\varepsilon_1-\varepsilon_2+\varepsilon_3, a-\varepsilon_1+\varepsilon_2, -a+\varepsilon_1, a) \mid \\ \varepsilon_1,\varepsilon_2,\varepsilon_3 \in \{-1,0,1\}, |a| \le 3t^2+9t+7\}.$$

Then we prove that it is a set of witnesses for $\mu_\gamma(X+t+1)$. Indeed (i) and (ii) of Theorem 2.1.4 obviously hold. We have

$$\frac{a(2+8t+6t^2+t^3) - (a-\varepsilon_1)(5+9t+3t^2) + (a-\varepsilon_1+\varepsilon_2)(4+3t) - a + \varepsilon_1 - \varepsilon_2 + \varepsilon_3}{1+2t+3t^2+t^3} = a + R,$$

where

$$R = \frac{\varepsilon_1(3t^2+6t+2) + 3\varepsilon_2(t+1) + \varepsilon_3 - a}{1+2t+3t^2+t^3}.$$

If $t > 6$ then $|R| < 1$. Thus, if $a \ge 0$, then $-\tilde{\tau}(e)_4 \le e_4$. If $a < 0$, then $\tilde{\tau}(e)_4 \le -e_4 + 1$ and if $a < -(3t^2+9t+6)$ then $\tilde{\tau}(e)_4 \le -e_4$, i.e. E satisfies (iii) too, as $t = 4, 5$ can be directly checked.

If $a < 0$ then $\tilde{\tau}(e)_4 \ge 0$. If $a > 0$ then applying $\tilde{\tau}$ some times we get $0 \le \tilde{\tau}(e)_4^k < a$. This shows that (iv) holds too, i.e. $\mu_\gamma(X+t+1) \in \mathcal{C}$.

Case(2) $\gamma = \alpha + t\alpha^2 - \alpha^3$. We have

$$\mu_\gamma(X+t+3) = X^4 + (12+t)X^3 + (53+9t)X^2 + (102+26t)X + 73 + 24t,$$

which is a CNS polynomial by Corollary 2.4.5, provided $t \ge 10$. For $t < 10$ we prove the same directly. Further we have

$$\mu_\gamma(X+t+2) = X^4 + (t+8)X^3 + (6t+23)X^2 + (28+11t)X + 13 + 6t,$$

which belongs to $\mathcal{C}$ by Lemma 2.4.9.

Case (3) $\gamma = t\alpha + (t-1)\alpha^2 - \alpha^3$ is obvious by Proposition 2.1.15.

Case (4) $\gamma = t\alpha - (t+1)\alpha^2 + \alpha^3$. We have

$$\mu_\gamma(X+1) = X^4 + (2t+6)X^3 + (9t+17)X^2 + (13t+24)X + 6t + 13,$$

which is not a CNS polynomial by Lemma 2.4.8. The minimal polynomial of $\gamma - 2$ is

$$\mu_\gamma(X+2) = X^4 + (2t+10)X^3 + (15t+41)X^2 + (37t+80)X + 61 + 30t$$

which is a CNS polynomial. We can prove it with Theorem 2.1.7 and entry vectors: $r_1 = [\frac{-1}{20}, \frac{1}{60}, \frac{9}{20}, \frac{71}{60}]$, $r_2 = [\frac{1}{60}, \frac{1}{60}, \frac{9}{20}, \frac{71}{60}]$, $r_3 = [\frac{-1}{20}, \frac{1}{12}, \frac{9}{20}, \frac{71}{60}]$, $r_4 = [\frac{-1}{20}, \frac{1}{60}, \frac{31}{60}, \frac{71}{60}]$, $r_5 = [\frac{-1}{20}, \frac{1}{60}, \frac{9}{20}, \frac{5}{4}]$.

Case(5) $\gamma = \alpha - \alpha^3$. We have

$$\begin{aligned}\mu_\gamma(X+1) &= X^4 + (t^3 - t + 4)X^3 + (3t^3 + 3t^2 - 3t + 5)X^2 + \\ &\quad (3t^3 + 6t^2 + 2)X + t^3 + 3t^2 + 2t + 1,\end{aligned}$$

which is not a CNS polynomial by Lemma 2.4.8. The minimal polynomial of $\gamma - 2$ is

$$\begin{aligned}\mu_\gamma(X+2) &= X^4 + (t^3 - t + 8)X^3 + (6t^3 + 3t^2 - 6t + 23)X^2 + \\ &\quad +(12t^3 + 12t^2 - 9t + 28)X + 8t^3 + 12t^2 - 2t + 13,\end{aligned}$$

for which we can apply Theorem 2.1.7 with entry vectors:

$r_1 = [\frac{-1}{48}, \frac{5}{48}, \frac{35}{48}, \frac{71}{48}]$, $r_2 = [\frac{1}{16}, \frac{5}{48}, \frac{35}{48}, \frac{71}{48}]$, $r_3 = [\frac{-1}{48}, \frac{3}{16}, \frac{35}{48}, \frac{71}{48}]$,
$r_4 = [\frac{-1}{48}, \frac{5}{48}, \frac{13}{16}, \frac{71}{48}]$, $r_5 = [\frac{-1}{48}, \frac{5}{48}, \frac{35}{48}, \frac{25}{16}]$.

Hence the polynomial is a CNS polynomial.

Case(6) $\gamma = \alpha - t(t^2+1)\alpha^2 + t^2\alpha^3$. As $k_\gamma = M(\gamma) + 2$, thus the proof is obvious by Proposition 2.1.15.

Cases (7) and (8) can be verified by direct computation. □

Chapter 3

Symmetric Shift Radix Systems

In this chapter first we define symmetric shift radix systems and some related number systems (symmetric β-expansions, symmetric canonical number systems). Then we list basic properties and the algorithm ([9]) applied to characterize symmetric shift radix systems. Two dimensional SSRS is treated in [9] by Akiyama and Scheicher, in the second and third sections we will show that $\mathcal{D}_3^0$ is the union of four polyhedra and a polygon.

The results of this chapter are based on [40], that is a joint work with Klaus Scheicher, Paul Surer and Jörg M. Thuswaldner.

We recall the definition of SRS: Let $d \geq 1$ be an integer, $\mathbf{r} \in \mathbb{R}^d$, and let

$$\tilde{\tau}_{\mathbf{r}} : \mathbb{Z}^d \to \mathbb{Z}^d, \quad \mathbf{a} = (a_1, \ldots, a_d) \mapsto (a_2, \ldots, a_d, -\lfloor \mathbf{ra} \rfloor),$$

where $\mathbf{ra} = r_1 a_1 + r_2 a_2 + \cdots + r_d a_d$, *i.e.*, the inner product of the vectors $\mathbf{r}$ and $\mathbf{a}$. Then $\tilde{\tau}_{\mathbf{r}}$ is called a *shift radix system* (for short SRS), if

$$\forall \mathbf{a} \in \mathbb{Z}^d \quad \exists n \in \mathbb{N} : \tilde{\tau}_{\mathbf{r}}^n(\mathbf{a}) = \mathbf{0}.$$

We will deal with a variant of SRS, the so-called symmetric shift radix system.

3.0.10 Definition. (***cf.*** **[9])** Let $d \geq 1$ be an integer, $\mathbf{r} \in \mathbb{R}^d$, and let

$$\tau_{\mathbf{r}} : \mathbb{Z}^d \to \mathbb{Z}^d, \quad \mathbf{a} = (a_1, \ldots, a_d) \mapsto \left(a_2, \ldots, a_d, -\left\lfloor \mathbf{ra} + \frac{1}{2} \right\rfloor \right). \qquad (3.0.1)$$

Then $\tau_{\mathbf{r}}$ is called a *symmetric shift radix system* (SSRS for short), if

$$\forall \mathbf{a} \in \mathbb{Z}^d \quad \exists n \in \mathbb{N} : \tau_{\mathbf{r}}^n(\mathbf{a}) = \mathbf{0}.$$

Observe that the only difference between the two definitions is just the additional summand "$+\frac{1}{2}$" inside the floor function in (6.0.1).

SSRS have been already treated by Akiyama and Scheicher [9]. It was proved there that, analogously to the classical SRS, we have a strong relationship to certain notions of number systems. In particular SSRS form a common generalization of symmetric β-expansions and symmetric canonical number systems (SCNS). For the sake of completeness we recall the definition of these number systems and summarize the results on their relation to SSRS.

3.0.11 Definition. (***cf.*** **[9])** Let $\beta > 1$ be a real non-integral number. The unique representation of a positive $\gamma \in \mathbb{R}$ of the form

$$\gamma = d_m \beta^m + d_{m-1}\beta^{m-1} + d_{m-2}\beta^{m-2} + \cdots$$

for some $m \in \mathbb{Z}$ with $d_k \in (-\frac{\beta+1}{2}, \ldots, \frac{\beta+1}{2}) \cap \mathbb{Z}$, $k \leq m$, such that the condition

$$-\frac{\beta^{k+1}}{2} \leq \sum_{i \leq k} d_i \beta^i < \frac{\beta^{k+1}}{2}$$

is satisfied for any $k \leq m$, is called the *symmetric β-expansion of γ*. We say that β has property (SF) if all $\gamma \in \mathbb{Z}[\beta^{-1}]$ admit a finite symmetric β-expansion.

In the same way as for property (F) of ordinary β-expansions (see [27]) it can be shown that a number β with property (SF) is necessarily a Pisot number.

3.0.12 Theorem. *A Pisot number β with minimal polynomial $(x - \beta)(x^{d-1} + r_{d-1}x^{d-2} + \cdots + r_2 x + r_1)$ has Property (SF) if and only if $\tau_{(r_1,\ldots,r_{d-1})}$ is an SSRS.*

There is a similar statement for SCNS whose definition we want to recall now.

3.0.13 Definition. (*cf.* [9]) Let $P(X) = X^d + p_{d-1}X^{d-1} + \cdots + p_1X + p_0 \in \mathbb{Z}[X]$, $|p_0| \geq 2$, $\mathcal{R} := \mathbb{Z}[X]/P(x)\mathbb{Z}[X]$, $x \in \mathcal{R}$ the image of X under the canonical epimorphism from $\mathbb{Z}[X]$ to $\mathcal{R}$ and $\mathcal{N} := \left[-\frac{|p_0|}{2}, \frac{|p_0|}{2}\right) \cap \mathbb{Z}$. $(P(X), \mathcal{N})$ is called a *symmetric canonical number system* (SCNS) if each $R \in \mathcal{R}$ can be represented as

$$R = \sum_{i=0}^{l} a_i x^i, \quad a_i \in \mathcal{N}.$$

3.0.14 Theorem. *$(P(X), \mathcal{N})$ with $P(X) = X^d + p_{d-1}X^{d-1} + \cdots + p_1X + p_0 \in \mathbb{Z}[X]$ and $\mathcal{N} := \left[-\frac{|p_0|}{2}, \frac{|p_0|}{2}\right) \cap \mathbb{Z}$ is an SCNS if and only if $\tau_{\mathbf{r}}$ is an SSRS, where $\mathbf{r} = \left(\frac{1}{p_0}, \frac{p_{d-1}}{p_0}, \ldots, \frac{p_1}{p_0}\right)$.*

3.1 Basic properties and algorithms for Symmetric Shift Radix Systems

Now, in order to show the differences between SSRS and SRS, we define the sets of $\mathcal{D}_d$, $\mathcal{D}_d^0$ related to the behavior of the periods of $\tau_{\mathbf{r}}$. Let

$$\mathcal{D}_d := \left\{ \mathbf{r} \in \mathbb{R}^d \,\middle|\, \forall \mathbf{a} \in \mathbb{Z}^d \, \exists n, l \in \mathbb{N} : \tau_{\mathbf{r}}^k(\mathbf{a}) = \tau_{\mathbf{r}}^{k+l}(\mathbf{a}) \ \forall k \geq n \right\} \text{ and}$$
$$\mathcal{D}_d^0 := \left\{ \mathbf{r} \in \mathbb{R}^d \,|\, \tau_{\mathbf{r}} \text{ is an SSRS} \right\}.$$

We recall that

$$\tilde{\mathcal{D}}_d = \left\{ \mathbf{r} \in \mathbb{R}^d \,\middle|\, \forall \mathbf{a} \in \mathbb{Z}^d \, \exists n, l \in \mathbb{N} : \tilde{\tau}_{\mathbf{r}}^k(\mathbf{a}) = \tilde{\tau}_{\mathbf{r}}^{k+l}(\mathbf{a}) \ \forall k \geq n \right\} \text{ and}$$
$$\tilde{\mathcal{D}}_d^0 = \left\{ \mathbf{r} \in \mathbb{R}^d \,|\, \tilde{\tau}_{\mathbf{r}} \text{ is an SRS} \right\},$$

For $\mathbf{r} = (r_1, \ldots, r_d) \in \mathbb{R}^d$, let

$$R(\mathbf{r}) = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ -r_1 & -r_2 & \cdots & -r_{d-1} & -r_d \end{pmatrix}.$$

For $M \in \mathbb{R}^{d\times d}$, denote by $\varrho(M)$ the spectral radius of M, *i.e.*, the maximum absolute value of the eigenvalues of M. For simplicity, we write $\varrho(\mathbf{r}) := \varrho(R(\mathbf{r}))$. Let

$$\mathcal{E}_d(\varepsilon) = \{\mathbf{r} \in \mathbb{R}^d : \varrho(\mathbf{r}) < \varepsilon\}.$$

It is known that the $\overline{\mathcal{E}_d(\varepsilon)}$ is a regular set, *i.e.*, the set coincides with the closure of its interior.

We start with the comparison of the sets $\mathcal{D}_d$ and $\tilde{\mathcal{D}}_d$. Firstly, it can easily be seen that their interiors are the same since from [2] we know $\mathcal{E}_d(1) \subset \tilde{\mathcal{D}}_d \subset \overline{\mathcal{E}_d(1)}$ while in [9] it has been shown that

$$\mathcal{E}_d(1) \subset \mathcal{D}_d \subset \overline{\mathcal{E}_d(1)}. \tag{3.1.1}$$

We will dwell upon the set $\mathcal{D}_d$ in Section 3.2. However, the sets $\mathcal{D}_d^0$ and $\tilde{\mathcal{D}}_d^0$ have different behavior. Properties of the set $\tilde{\mathcal{D}}_d^0$ have been developed in [2, 3, 4]. In [3, 83] special attention was paid to the two dimensional case $\tilde{\mathcal{D}}_2^0$. It turns out that the structure of $\tilde{\mathcal{D}}_2^0$ is very complicated and although large parts of the set could be characterized, a full characterization is still outstanding. An approximation of $\tilde{\mathcal{D}}_2^0$ is shown in Figure 3.1.

The sets $\tilde{\mathcal{D}}_d^0$ for $d \geq 3$ are not yet investigated in detail, however, computer experiments indicate that $\tilde{\mathcal{D}}_3^0$ is hard to describe.

For the case of SSRS the situation becomes more pleasant at least for low dimensions. Akiyama and Scheicher [9] presented that $\mathcal{D}_2^0$ has a simple characterization (see Figure 3.2). They found out that

$$\mathcal{D}_2^0 = \left\{(x,y) \in \mathbb{R}^2 \;\middle|\; x \leq \frac{1}{2}, -x - \frac{1}{2} < y \leq x + \frac{1}{2}\right\} \setminus \left\{(\frac{1}{2}, y) \in \mathbb{R}^2 \;\middle|\; \frac{1}{2} < y < 1\right\},$$

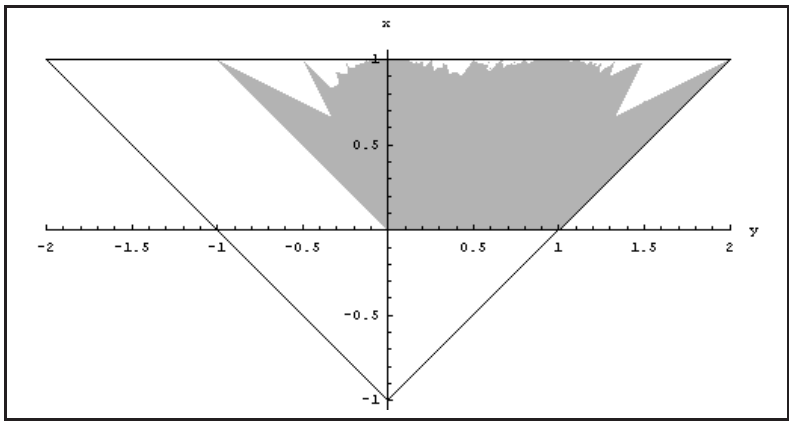

Figure 3.1: An approximation of $\tilde{\mathcal{D}}_2^0$

i.e., $\mathcal{D}_2^0$ is an isosceles triangle together with some parts of its boundary. We are interested in the shape of the set $\mathcal{D}_3^0$.

Let us consider the set $\mathcal{D}_d$. By (6.0.4) apart from the boundary, the set $\mathcal{D}_d$ coincides with the set $\mathcal{E}_d(1)$ and their closures are equal. As the minimal polynomial of $R(\mathbf{r})$ is given by

$$x^d + r_d x^{d-1} + \cdots + r_2 x + r_1 \tag{3.1.2}$$

the problem of characterizing $\mathcal{E}_d(\varepsilon)$ is equivalent to the problem of finding polynomials of the form (3.1.2) whose roots lie inside the ε multiple of the unit ball. This problem was already settled in [78, 84]. From these references we easily get the following lemma.

3.1.1 Lemma. *A vector $\boldsymbol{r} = (r_1, ..., r_d)$ is contained in $\mathcal{E}_d(\varepsilon)$ if and only if the Hermitian form*

$$H_d(x_0, \ldots, x_{d-1}) := \sum_{i=0}^{d-1} \left| \sum_{j=i}^{d-1} \varepsilon^{d+i-j} r_{d+i-j+1} x_j \right|^2 - \sum_{i=0}^{d-1} \left| \sum_{j=i}^{d-1} \varepsilon^{j-i} r_{j-i+1} x_j \right|^2$$

with $r_{d+1} = 1$ is positive definite.

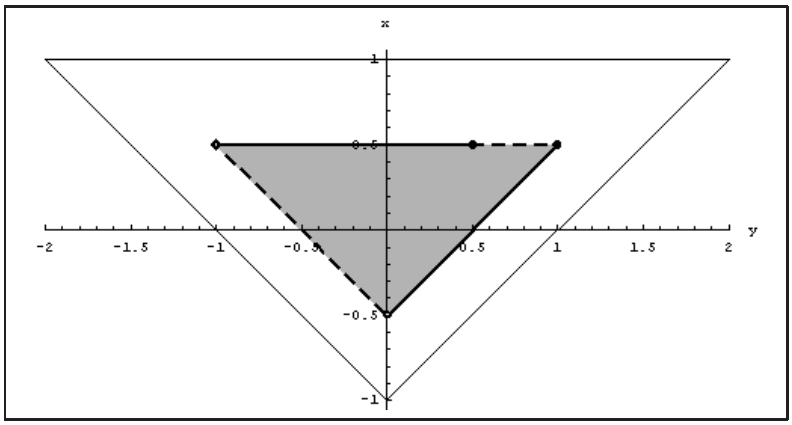

Figure 3.2: The shape of $\mathcal{D}_2^0$

Now we turn to the study of $\mathcal{D}_d^0$. The set $\mathcal{D}_d^0$ can be constructed from the set $\mathcal{D}_d$ by cutting out convex polyhedra. For $\mathbf{r} = (r_1, \ldots, r_d) \in \mathcal{D}_d$, an element $\mathbf{a} = (a_1, \ldots, a_d) \in \mathbb{Z}^d \setminus \{0\}$ is a *non-zero periodic point of* $\tau_{\boldsymbol{r}}$ *of period* L, if $\mathbf{a} = \tau_{\mathbf{r}}^L(\mathbf{a})$. From the definition of $\mathcal{D}_d^0$ it follows that the existence of such a periodic point is necessary and sufficient for $\mathbf{r} \notin \mathcal{D}_d^0$. Suppose that the period defined by $\mathbf{a}$ runs through the orbit

$$\tau_{\mathbf{r}}^j(\mathbf{a}) = (a_{1+j}, \ldots, a_{d+j}) \quad (0 \le j \le L-1),$$

where $a_{L+1} = a_1, ..., a_{L+d-1} = a_{d-1}$. We denote such a period by

$$(a_1, \ldots, a_d); a_{d+1}, \ldots, a_L$$

and say that it is a period of $\tau_{\mathbf{r}}$ or just a *period of* $\mathcal{D}_d$.

Let a non-zero period $\pi := (a_1, \ldots, a_d); a_{d+1}, \ldots, a_L$ be given. We may ask for the set $P(\pi)$ of all $\mathbf{r} \in \mathcal{D}_d$ for that π occurs as a period of $\tau_{\mathbf{r}}$. By the definition of $\tau_{\mathbf{r}}$, an element $\mathbf{r} \in P(\pi)$ has to satisfy the system of L double inequalities

$$-\frac{1}{2} \le r_1 a_{1+i} + r_2 a_{2+i} + \cdots + r_d a_{d+i} + a_{d+1+i} < \frac{1}{2}. \tag{3.1.3}$$

Here i runs from 0 to $L-1$ and $a_{L+1} = a_1, \ldots, a_{L+d} = a_d$. Such a system characterizes a convex polyhedron, which is possibly degenerated or

equal to the empty set. Therefore we will call $P(\pi)$ a *cutout polyhedron*. Example 3.2.1 shows how $P(\pi)$ could look like for a given period in the three dimensional case. Since each point $\mathbf{r} \in P(\pi)$ has π as a period of the associated mapping $\tau_{\mathbf{r}}$ the set $P(\pi)$ has empty intersection with $\mathcal{D}_d^0$. Thus we get the representation

$$\mathcal{D}_d^0 = \mathcal{D}_d \setminus \bigcup_{\pi \neq \mathbf{0}} P(\pi),$$

where the union is extended over all non-zero periods π. Since the set of periods is infinite, this expression is not suitable for calculations. The following theorem shows how to reduce the set of possible periods to a finite set and gives an efficient algorithm for a closed subset H of $\operatorname{int} \mathcal{D}_d = \mathcal{E}_d(1)$ to determine $H \cap \mathcal{D}_d^0$. It was presented for the first time for CNS in [15] and further improved and adapted to SRS in [2, 3, 83]. In [9] the algorithm was established for SSRS. Basically we will use this version. Let $\mathbf{e}_i$ be the i-th canonical unit vector. For an $\mathbf{r} = (r_1, \ldots, r_d) \in \operatorname{int} \mathcal{D}_d$, denote by $\mathcal{V}(\mathbf{r}) \subset \mathbb{Z}^d$ the smallest set with the properties

1. $\pm \mathbf{e}_i \in \mathcal{V}(\mathbf{r}), i = 1, \ldots, d$,
2. $(a_1, \ldots, a_d) \in \mathcal{V}(\mathbf{r}) \Rightarrow (a_2, \ldots, a_{d+1}) \in \mathcal{V}(\mathbf{r})$ where a_{d+1} satisfies
$$-1 < r_1 a_1 + r_2 a_2 + \cdots + r_d a_d + a_{d+1} < 1.$$

$\mathcal{V}(\mathbf{r}) \subset \mathbb{Z}^d$ is called a *set of witnesses* for $\mathbf{r}$. Additionally define $\mathcal{G}(\mathcal{V}(\mathbf{r})) = V \times E$ to be the graph with set of vertices $V = \mathcal{V}(\mathbf{r})$ and set of edges $E \subset V \times V$ such that

$$\forall \mathbf{a} \in V : (\mathbf{a}, \tau_{\mathbf{r}}(\mathbf{a})) \in E.$$

The set of vertices is exactly the same as in [2]. The edges are defined in a different way. There exists only one outgoing edge for each vertex. We are interested in the cyclic structure of such graphs. A cycle $\mathbf{a}_1 \to \mathbf{a}_2 \to \cdots \to \mathbf{a}_L \to \mathbf{a}_1$ induces a periodic point of period L in an obvious way.

3.1.2 Theorem. (*cf.* [9]) *Let $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k \in \mathcal{D}_d$ and let $H := \square(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k)$ be the convex hull of $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k$. Assume that $H \subset \operatorname{int} \mathcal{D}_d$ and sufficiently small in diameter. Then there exists an algorithm to construct a finite directed graph $G(H) = V \times E$ with vertices $V \subset \mathbb{Z}^d$ and edges $E \subset V \times V$ which satisfies*

1. $\pm \boldsymbol{e}_i \in V$ *for all* $i = 1, \ldots, d$,

2. $\mathcal{G}(\mathcal{V}(\boldsymbol{x}))$ *is a subgraph of* $G(H)$ *for all* $\boldsymbol{x} \in H$,

3. $H \cap \mathcal{D}_d^0 = H \setminus \bigcup_\pi P(\pi)$, *where* π *runs through all periods induced by the nonzero primitive cycles of* G.

Observe that the theorem can be extended to any convex set $H \subset \operatorname{int} \mathcal{D}_d$ analogously to [83]. In our context the version presented in Theorem 6. suffices. In practice, the graph in Theorem 6. is constructed by successively adding new vertices. Note that the restriction "sufficiently small" is not superfluous. It turns out that the size of the set of vertices in the graph in Theorem 6. can grow to infinity if H is chosen too big. For more detailed information on this topic, see [9, 83]. For us it is only important to choose H in a way that everything stays finite. This can be realized by a suitable subdivision of a given set. We will turn to this problem in Section 3.3.

Theorem 6. proved to be a powerful tool for characterizing $\mathcal{D}_d^0$. If it is used properly, $\mathcal{D}_d^0 \cap H$ can be characterized for any closed $H \subset \operatorname{int} \mathcal{D}_d$. Thus, whenever there exists such an H with $\mathcal{D}_d^0 \subset H$ there is a chance to characterize $\mathcal{D}_d^0$ completely. That was the case for $d = 2$ and we will see that this is valid for $d = 3$, too. For classical SRS, there does not exist such a set H for $d \geq 2$.

3.2 Construction of $\mathcal{D}_3^0$ from $\mathcal{D}_3$

Our aim is to characterize $\mathcal{D}_3^0$. We already know that

$$\mathcal{E}_3(1) \subset \mathcal{D}_3 \subset \overline{\mathcal{E}_3(1)}.$$

From Lemma 3.1.1 we calculate

$$\mathcal{E}_3(1) = \left\{ (x, y, z) \in \mathbb{R}^3 \;\middle|\; |x| < 1, |y - xz| < 1 - x^2, |x + z| < |y + 1| \right\}.$$

The following example shows how a given period cuts out a polyhedron from $\mathcal{E}_3(1)$.

3.2.1 Example. Consider the period $\pi := (1,1,-1);-1,0$. It induces a system of inequalities (6.0.5) which describes the polyhedron $P(\pi)$. In our case we get

$$\begin{aligned} P(\pi) = \Big\{(x,y,z)\Big| & -\frac{1}{2} \le x+y-z-1 < \frac{1}{2} \wedge -\frac{1}{2} \le x-y-z < \frac{1}{2} \\ & \wedge -\frac{1}{2} \le -x-y+1 < \frac{1}{2} \wedge -\frac{1}{2} \le -x+z+1 < \frac{1}{2} \\ & \wedge -\frac{1}{2} \le y+z-1 < \frac{1}{2}\Big\}. \end{aligned}$$

By removing redundant inequalities, this reduces to

$$\begin{aligned} P(\pi) = \Big\{(x,y,z)\Big| \; & x+y-z-1 < \frac{1}{2} \wedge x-y-z < \frac{1}{2} \wedge -\frac{1}{2} \le -x-y+1 \\ & \wedge -x+z+1 < \frac{1}{2} \wedge -\frac{1}{2} \le y+z-1\Big\} \end{aligned}$$

yielding a polyhedron with five faces. $P(\pi)$ only contains $\mathbf{r} \in \mathcal{D}_d$ with $\tau_{\mathbf{r}}^5((1,1,-1)) = (1,1,-1)$ and, hence, $P(\pi)$ has empty intersection with $\mathcal{D}_3^0$. Figure 3.3 shows the position of $P(\pi)$ in $\mathcal{E}_3(1)$. It is easy to see that $P(\pi)$ really cuts out something.

In the sequel we will need $\overline{\mathcal{E}_3(1)}$ and there some problems occur. Suppose the set which is obtained by changing all the strict inequalities ("$<$") of $\mathcal{E}_3(1)$ to non strict inequalities ("$\le$"). One may think that it equals $\overline{\mathcal{E}_3(1)}$, but this is not the case. It can be easily seen that this set contains the unbounded lines $(1,\lambda,\lambda), \lambda \in \mathbb{R}$ and $(-1,\mu,-\mu), \mu \in \mathbb{R}$ which cannot be true for $\overline{\mathcal{E}_3(1)}$. Hence, $\overline{\mathcal{E}_3(1)}$ is only a subset of this set. We will solve the problem by adding some suitable inequalities. Let

$$\begin{aligned} \mathcal{E}_3' := \big\{(x,y,z) \in \mathbb{R}^3\big| \; & |x| \le 1 \wedge |y-xz| \le 1-x^2 \\ & \wedge |x+z| \le |y+1| \wedge |y-1| \le 2 \wedge |z| \le 3\big\} \end{aligned}$$

and consider the intersection of $\mathcal{E}_3'$ with the hyperplane

$$A_c := \big\{(x,y,z) \in \mathbb{R}^3 \mid x-c=0\big\}$$

for constant c.

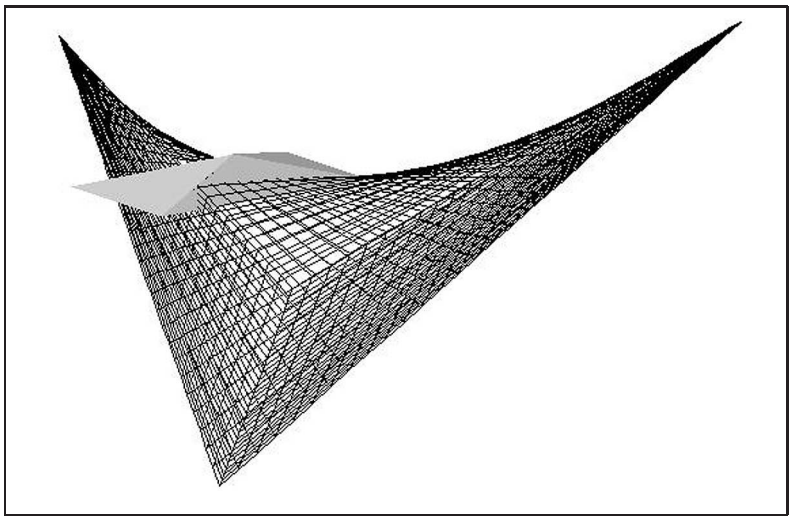

Figure 3.3: The position of $P(\pi)$ in $\mathcal{E}_3(1)$

3.2.2 Lemma. *For any $|c| < 1$ the intersection of $\mathcal{E}'_3$ with the plane A_c yields the closed triangle $\triangle(A_c^{(1)}, A_c^{(2)}, A_c^{(3)})$ with $A_c^{(1)} = (c, -1, -c), A_c^{(2)} = (c, 1-2c, c-2), A_c^{(3)} = (c, 2c+1, c+2)$.*

Proof We have

$$\mathcal{E}'_3 \cap A_c = \big\{(c,y,z) \in \mathbb{R}^3 \big| \; |y - cz| \le 1 - c^2 \wedge |c + z| \le |y + 1| \\ \wedge |y - 1| \le 2 \wedge |z| \le 3\big\}.$$

As all inequalities are linear, this is a convex set. It is quickly verified that $A_c^{(1)}, A_c^{(2)}, A_c^{(3)} \in \mathcal{E}'_3 \cap A_c$. Thus $\triangle(A_c^{(1)}, A_c^{(2)}, A_c^{(3)}) \subset \mathcal{E}'_3 \cap A_c$. On the other hand consider the closed convex set

$$B_c := \big\{(c,y,z) \;\big|\; y - cz \le 1 - c^2 \wedge c + z \le y + 1 \wedge -y - 1 \le c + z\big\}.$$

Observe that for its definition we used only inequalities that occurred in the definition of $\mathcal{E}'_3 \cap A_c$ and hence we have $\mathcal{E}'_3 \cap A_c \subset B_c$. Pairwise intersection of the three boundary lines of B_c yields exactly the three points $A_c^{(1)}, A_c^{(2)}, A_c^{(3)}$ and therefore $\triangle(A_c^{(1)}, A_c^{(2)}, A_c^{(3)}) = B_c \supset \mathcal{E}'_3 \cap A_c$. $\square$

3.2.3 Theorem. $\overline{\mathcal{E}_3(1)} = \mathcal{E}_3'$.

Proof Obviously $\mathcal{E}_3'$ is a closed set while $\mathcal{E}_3(1)$ is open. We state that $\operatorname{int}\mathcal{E}_3' = \mathcal{E}_3(1)$. From Lemma 2. we know

$$\mathcal{E}_3' \cap A_c = \{(c,y,z) \mid y - cz \leq 1 - c^2 \wedge c + z \leq y + 1 \wedge -y - 1 \leq c + z\}$$

and as every point of $\mathcal{E}_3(1)$ is inside $\mathcal{E}_3' \cap A_c$ for some $|c| < 1$ we have

$$\mathcal{E}_3' = \bigcup_{|c|\leq 1} (\mathcal{E}_3' \cap A_c) \supset \mathcal{E}_3(1)$$

and therefore

$$\operatorname{int}\mathcal{E}_3' \supset \operatorname{int}\mathcal{E}_3(1) = \mathcal{E}_3(1).$$

On the other hand denote by $\operatorname{int}_{A_c}(\mathcal{E}_3' \cap A_c)$ the interior of the set $\mathcal{E}_3' \cap A_c$(subspace topology) for $|c| < 1$, *i.e.*, the open triangle defined in Lemma 2., and observe that

$$\operatorname{int}\mathcal{E}_3' = \bigcup_{|c|<1} \operatorname{int}_{A_c}(\mathcal{E}_3' \cap A_c)$$

as we can find a neighborhood around each point of $\operatorname{int}_{A_c}(\mathcal{E}_3' \cap A_c), |c| < 1$ which is contained in $\mathcal{E}_3'$. Further each point of $\operatorname{int}_{A_c}(\mathcal{E}_3' \cap A_c)$ satisfies the conditions of $\mathcal{E}_3(1)$ whenever $|c| < 1$. Hence

$$\operatorname{int}\mathcal{E}_3' = \bigcup_{|c|<1} \operatorname{int}(\mathcal{E}_3' \cap A_c) \subset \mathcal{E}_3(1).$$

Thus we have shown that $\operatorname{int}\mathcal{E}_3' = \mathcal{E}_3(1)$.

To prove the theorem we show $\mathcal{E}_3' = \overline{\operatorname{int}\mathcal{E}_3'}$. We already have that $\operatorname{int}\mathcal{E}_3' = \bigcup_{|c|<1} \operatorname{int}_{A_c}(\mathcal{E}_3' \cap A_c)$. Hence we look at the convergent sequences of points contained in $\bigcup_{|c|<1} \operatorname{int}(\mathcal{E}_3' \cap A_c)$. Such a sequence converges either to some point within $\bigcup_{|c|<1}(\mathcal{E}_3' \cap A_c)$ or to some point within one of the sets $\lim_{c\to\pm 1}(\mathcal{E}_3' \cap A_c)$. From Lemma 2. we already have

$$\mathcal{E}_3' \cap A_c = \triangle((c,-1,-c)(c,1-2c,c-2),(c,2c+1,c+2))$$

and we see that

$$\lim_{c\to 1}(\mathcal{E}_3' \cap A_c) = \{(1,\lambda,\lambda) \mid -1 \le \lambda \le 3\},$$
$$\lim_{c\to -1}(\mathcal{E}_3' \cap A_c) = \{(-1,\lambda,-\lambda) \mid -1 \le \lambda \le 3\}$$

which exactly correspond to the sets $(\mathcal{E}_3' \cap A_{\pm 1})$. Thus

$$\overline{\mathcal{E}_3(1)} = \overline{\operatorname{int} \mathcal{E}_3'} = \bigcup_{|c|\le 1} (\mathcal{E}_3' \cap A_c) = \mathcal{E}_3'$$

and we are done. □

Finally we have a representation of the closure of $\mathcal{E}_3(1)$. In the proof of Lemma 2. we already recognized that the number of inequalities to describe $\overline{\mathcal{E}_3'}$ can be reduced. Indeed, by using an algorithm (Algorithm 3) which we will present in Section 3.3, we gain

$$\overline{\mathcal{E}_3(1)} = \left\{(x,y,z) \,\middle|\, |x+z| \le 1 + y \wedge y - xz \le 1 - x^2 \wedge |z| \le 3 \right\}.$$

3.3 Characterization of $\mathcal{D}_3^0$

In this section we give a complete description of $\mathcal{D}_3^0$. For this reason we define the sets

$$\begin{aligned}
S_1 := \{(x,y,z) \mid\ & 2x - 2z \ge 1 \wedge 2x + 2y + 2z > -1 \wedge 2x + 2y \le 1 \\
& \wedge 2x \le 1 \wedge 2x - 2y + 2z \le 1\},\\
S_2 := \{(x,y,z) \mid\ & x - z \le -1 \wedge 2x - 2y + 2z \le 1 \wedge -2x + 2y \le 1 \\
& \wedge 2x > -1\},\\
S_3 := \{(x,y,z) \mid\ & x - z > -1 \wedge 2x - 2y + 2z \le 1 \wedge -2x + 2y < 1, 2x > -1 \\
& \wedge 2x - 2z < -1 \wedge 2x + 2y + 2z > -1\},\\
S_4 := \{(x,y,z) \mid\ & 2x - 2y + 2z \le 1 \wedge -2x + 2y \le 1 \wedge 2x - 2z = -1 \\
& \wedge 2x + 2y + 2z > -1\},\\
S_5 := \{(x,y,z) \mid\ & -1 < 2x \le 1 \wedge -1 < 2x - 2z \le 1 \wedge 2x + 2y + 2z > -1 \\
& \wedge 2x - 2y + 2z \le 1 \wedge 2x + 4y - 2z < 3, 2y \le 1\}
\end{aligned}$$

and denote their union by

$$\mathcal{S} := \bigcup_{i\in\{1,\dots,5\}} S_i.$$

Note that S_1, S_2, S_3, S_5 are polyhedra while S_4 is a polygon. The following theorem states the main result.

3.3.1 Theorem. $\mathcal{D}_3^0 = \mathcal{S}$

Two views of the set $\mathcal{D}_3^0$ are depicted in Figure 5.1 and Figure 3.5. For rotating 3D-pictures of $\mathcal{D}_3^0$ we refer the reader to the authors' home pages [82].

In subsection 3.3 we will prove this theorem. Here we want to give an outline of the proof. In a first step we will use Theorem 6. in order to show that

$$\mathcal{S} \subseteq \mathcal{D}_3^0. \tag{3.3.1}$$

For showing the opposite inclusion we need a set of nonzero periods Π such that for $\mathcal{P} := \bigcup_{\pi\in\Pi} P(\pi)$ we have

$$\mathcal{S} \cup \mathcal{P} \supseteq \mathcal{D}_3.$$

From (6.0.7) we can deduce $\mathcal{S} \cap \mathcal{P} = \emptyset$. Thus,

$$\mathcal{S} \supseteq \mathcal{D}_3 \setminus \mathcal{P} \supseteq \mathcal{D}_3^0.$$

Since $\mathcal{D}_3 \subset \overline{\mathcal{E}_3(1)}$ we are done if we can cover $\overline{\mathcal{E}_3(1)}$ with $\mathcal{P} \cup \mathcal{S}$, *i.e.*, if we can show that

$$\mathcal{P} \cup \mathcal{S} \supseteq \overline{\mathcal{E}_3(1)}.$$

Proof of the main result

We will prove our result in two parts according to the outline given in the previous section. First of all, we set up some notation.

3.3.2 Notation For a logical system $\mathcal{J}$ of inequalities, which are combined by $\wedge$ and $\vee$, denote by $X(\mathcal{J})$ the set of all points that satisfy $\mathcal{J}$. Let P a set of inequalities. Then $\bigwedge P$ and $\bigvee P$ denote the systems $\bigwedge_{I\in P} I$ and $\bigvee_{I\in P} I$, respectively.

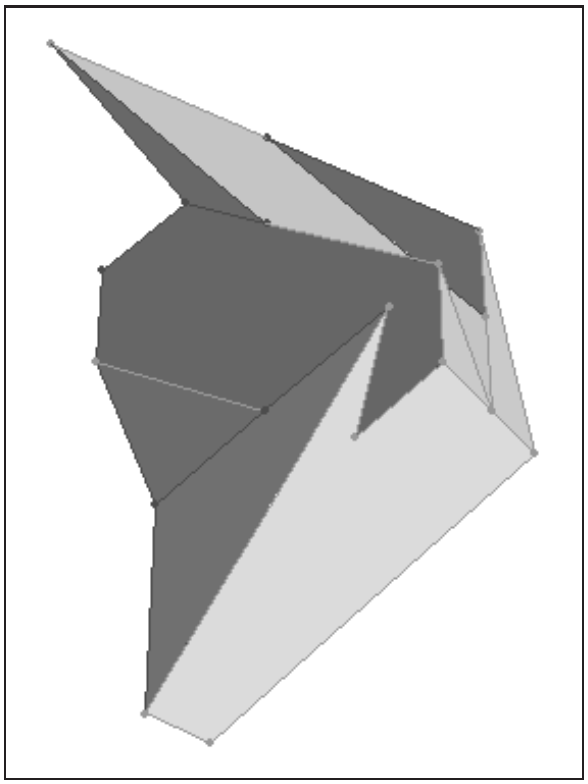

Figure 3.4: A view of $\mathcal{D}_3^0$

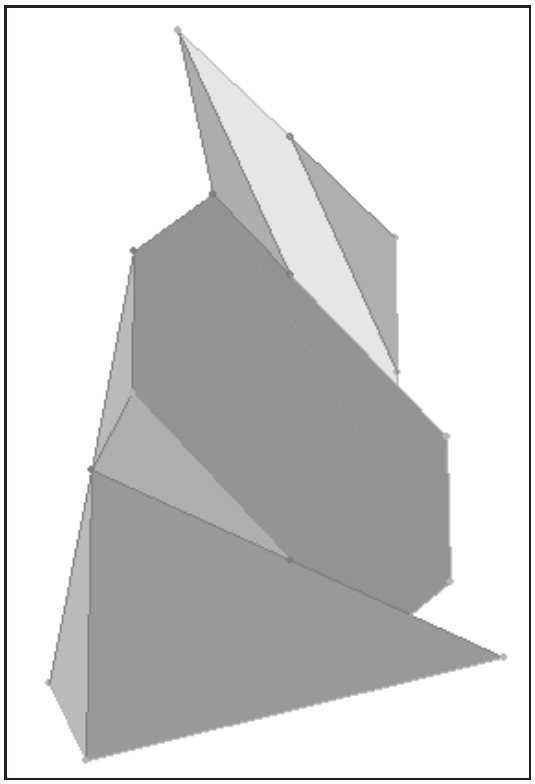

Figure 3.5: A view of $\mathcal{D}_3^0$

Algorithm 2 is recursive. For the rest of the section denote by T_i the set of inequalities that define the set S_i for $i \in \{1, \dots, 5\}$. These sets are assembled only of single inequalities. We have

$$\begin{aligned}
T_1 &:= \{2x - 2z \geq 1, 2x + 2y + 2z > -1, 2x + 2y \leq 1, 2x \leq 1,\\
&\qquad 2x - 2y + 2z \leq 1\},\\
T_2 &:= \{x - z \leq -1, 2x - 2y + 2z \leq 1, -2x + 2y \leq 1, 2x > -1\},\\
T_3 &:= \{x - z > -1, 2x - 2y + 2z \leq 1, -2x + 2y < 1, 2x > -1,\\
&\qquad 2x - 2z < -1, 2x + 2y + 2z > -1\},\\
T_4 &:= \{2x - 2y + 2z \leq 1, -2x + 2y \leq 1, 2x - 2z \leq -1, 2x - 2z \geq -1,\\
&\qquad 2x + 2y + 2z > -1\},\\
T_5 &:= \{ -1 < 2x, 2x \leq 1, -1 < 2x - 2z, 2x - 2z \leq 1, 2x + 2y + 2z > -1,\\
&\qquad 2x - 2y + 2z \leq 1, 2x + 4y - 2z < 3, 2y \leq 1\},
\end{aligned}$$

hence the equality of S_4 and the two double inequalities of S_5 are split into inequalities. Thus, $S_i = X(\bigwedge T_i)$ for $i = 1, \dots, 5$. Denote by $\bar{T}_i$ the set T_i with all the strict inequalities changed to not strict ones. Since all occurring inequalities are linear it can easily be checked that $\overline{S_i} = X(\bigwedge \bar{T}_i)$.

Table 3.1 shows 43 different periods with corresponding period length L, we denote the corresponding polyhedron by $P(\pi_j)$, where $j \in \{1, \dots, 43\}$.

Now for each $i \in \{1, \dots, 43\}$ define Q_i as the set of single inequalities such that $P(\pi_i) = X(\bigwedge Q_i)$. For instance, the set Q_{19} can be defined by

$$\begin{aligned}
Q_{19} := \Big\{ &-\frac{1}{2} \leq x + y - z - 1, x + y - z - 1 < \frac{1}{2}, -\frac{1}{2} \leq x - y - z,\\
&x - y - z < \frac{1}{2}, -\frac{1}{2} \leq -x - y + 1, -x - y + 1 < \frac{1}{2},\\
&-\frac{1}{2} \leq -x + z + 1, -x + z + 1 < \frac{1}{2}, -\frac{1}{2} \leq y + z - 1,\\
&y + z - 1 < \frac{1}{2}\Big\}
\end{aligned}$$

(see also Example 3.2.1). Finally we set

$$\mathcal{P} := \bigcup_{j=1}^{43} P(\pi_j).$$

L	Periods
3	π_1=$(-1,-1,-1)$ π_2=$(-1,-1,0)$ π_3=$(-1,0,1)$ π_4=$(0,-1,0)$ π_5=$(0,-1,1)$
4	π_6=$(0,-1,0);-1$ π_7=$(0,-1,0);1$ π_8=$(1,-1,1);-1$
5	π_9=$(-2,1,-1);-1,1$ π_{10}=$(-2,1,0);-1,2$ π_{11}=$(-1,-1,1);1,0$ π_{12}=$(0,-2,-1);1,2$ π_{13}=$(0,-1,1);-1,0$ π_{14}=$(0,1,-1);1,0$ π_{15}=$(0,1,0);-1,-1$ π_{16}=$(0,1,0);-1,0$ π_{17}=$(0,2,1);-1,-2$ π_{18}=$(1,-1,1);-1,0$ π_{19}=$(1,1,-1);-1,0$ π_{20}=$(2,-1,0);1,-2$
6	π_{21}=$(0,-1,0);0,1,0$ π_{22}=$(1,1,0);-1,-1,0$
7	π_{23}=$(0,1,-1);-1,1,0,-1$ π_{24}=$(1,1,0);-1,-1,-1,0$
8	π_{25}=$(-1,-1,1);1,2,0,0,-2$ π_{26}=$(-1,0,0);1,0,0,-1,-1$ π_{27}=$(-1,1,0);-1,1,-1,0,1$ π_{28}=$(0,0,2);1,1,-1,-1,-2$ π_{29}=$(1,1,1);0,-1,-1,-1,0$ π_{30}=$(2,1,-1);-2,-2,-1,1,2$
9	π_{31}=$(-1,0,0);1,1,1,0,-1,-1$ π_{32}=$(0,1,1);1,0,-1,-2,-2,-1$
10	π_{33}=$(-1,-1,1);0,-1,1,1,-1,0,1$ π_{34}=$(0,-2,1);1,-2,0,2,-1,-1,2$ π_{35}=$(0,-1,-1);-1,0,0,1,1,1,0$ π_{36}=$(1,2,1);1,-1,-1,-2,-1,-1,1$ π_{37}=$(1,2,2);1,0,-1,-2,-2,-1,0$
11	π_{38}=$(-2,0,1);-2,1,0,-2,2,-1,-1,2$ π_{39}=$(0,1,2);2,1,0,-1,-2,-2,-2,-1$
12	π_{40}=$(-2,2,-1);0,1,-2,2,-2,1,0,-1,2$ π_{41}=$(0,1,2);2,2,1,0,-1,-2,-2,-2,-1$
13	π_{42}=$(0,1,-2);2,-1,-1,2,-2,1,0,-1,1,-1$
22	π_{43}=$(0,2,2);1,-1,-2,-2,0,1,2,1,0,$ $-2,-2,-1,1,2,2,0,-1,-2,-1$

Table 3.1: The 43 periods needed to cut out $\mathcal{D}_3^0$

3.3.3 Remark. We note that the construction of the set $\mathcal{S}$ as well as the exhibition of the 43 periods corresponding to relevant cutout polyhedra has been achieved by extensive computer experiments. Up to now we do not know an easy way that would lead to a list of all the cutouts needed to get the set $\mathcal{D}_3^0$. To find an algorithmic way to construct all these cutouts is desirable since it could lead to characterizations of $\mathcal{D}_d^0$ even for higher dimensions d.

Observe that no element of the 43 periods given above contains elements having modulus greater than 2. Up to now, we do not know the reason for this fact. In order to characterize $\tilde{\mathcal{D}}_2^0$ we need periods with elements that are arbitrarily large (*cf.* [2, Sections 6 and 7]).

Using the algorithm of section 3.2. Theorem 6. shows the existence of an algorithm for the construction of a graph $G(H) = V \times E$ which can be used for finding all periods within the convex body H. Following [9], the graph is constructed recursively. Define $H = \square(\mathbf{r}_1, ..., \mathbf{r}_k) \subset \operatorname{int} \mathcal{D}_3$ to be the convex hull of some points $\mathbf{r}_1, \ldots, \mathbf{r}_k$. For a $\mathbf{z} \in \mathbb{Z}^d$, let $m(\mathbf{z}) = \min_{i\in\{1,\ldots,k\}}(-\lfloor \mathbf{r}_i\mathbf{z} \rfloor)$ and $M(\mathbf{z}) = \max_{i\in\{1,\ldots,k\}}(-\lfloor \mathbf{r}_i\mathbf{z} \rfloor)$. Set

$$V_0 := \{\pm\mathbf{e}_i \mid i = 1, \ldots, d\}$$

and then successively calculate $V_1, V_2, \ldots$ by the rule

$$V_{i+1} := V_i \cup \{(z_2, \ldots, z_d, j) \mid \mathbf{z} = (z_1, \ldots, z_d) \in V_i, -M(-\mathbf{z}) \le j \le M(\mathbf{z})\}.$$

For sets H having a sufficiently small diameter the iteration stabilizes yielding $V := V_n = V_{n+1}$ for some $n \in \mathbb{N}$. The set of edges is constructed by

$$E := \{(\mathbf{x}, (z_2, \ldots, z_d, j)) \mid \mathbf{x} = (z_1, \ldots, z_d) \in V, m(\mathbf{z}) \le j \le M(\mathbf{z})\}.$$

Let $\mathcal{Q}$ be a system of linear, non-strict inequalities linked with $\wedge$. Then $X(\mathcal{Q})$ forms a convex polyhedron that can be regarded as the convex hull of finitely many points $\mathbf{r}_1, ..., \mathbf{r}_k$. Suppose that $X(\mathcal{Q}) \subset \mathcal{E}_3(1)$. We want to set up an algorithm that calculates the set of all periods π whose associated polyhedron $P(\pi)$ has non-empty intersection with $X(\mathcal{Q})$. Theorem 6. ensures the existence of such an algorithm only if $X(\mathcal{Q})$ has sufficiently small diameter. If the set $X(\mathcal{Q})$ is too big, the graph $G(X(\mathcal{Q}))$ is infinite. We solve this problem in the following way. Suppose that, during the calculation of $|V|$, we obtain a set V_i whose number of elements $|V_i|$ exceeds an

appropriate bound p. In this case we stop the calculation of V and divide the set $X(\mathcal{Q})$ into two parts for which we calculate the set V again. By recursively doing this splitting procedure we eventually end up with sets whose diameter is small enough (provided that p is chosen reasonably).

Suppose that the set $X(\mathcal{Q})$ is the convex hull of its k vertices $\mathbf{r}_1, \ldots, \mathbf{r}_k$. We do not know these vertices explicitly. What we need is just $m(\mathbf{z})$ and $M(\mathbf{z})$ for certain fixed values of $\mathbf{z} \in \mathbb{Z}^d$. However, as $\mathcal{Q}$ is given as a system of linear inequalities, we easily see that

$$\begin{aligned} m(z) &= \min_{\mathbf{r} \in X(\mathcal{Q})} (-\lfloor \mathbf{rz} \rfloor), \\ M(z) &= \max_{\mathbf{r} \in X(\mathcal{Q})} (-\lfloor \mathbf{rz} \rfloor). \end{aligned}$$

The extremal values on the left hand side can now easily be calculated by standard linear optimization.

The algorithm consists of two parts. The first part is Algorithm 1, which constructs the set of vertices V of the graph $G(X(\mathcal{Q}))$ for a given convex body $X(\mathcal{Q})$. Whenever during the calculation the size of this set exceeds a given bound p, Algorithm 1 stops returning an overflow. Otherwise it terminates by returning V. Denote the application of Algorithm 1 with parameter $\mathcal{Q}$ and bound p by $\mathrm{VG}(\mathcal{Q}, p)$ (VG = **v**ertices of the **g**raph).

Algorithm 2 is recursive. As input we have $\mathcal{Q}$ and we write $\mathrm{FP}(\mathcal{Q})$ for its application on $\mathcal{Q}$ (FP= **f**ind all **p**eriods). Algorithm 2 evokes Algorithm 1 to calculate the set of vertices of $G(X(\mathcal{Q}))$. If an overflow occurs, the set $X(\mathcal{Q})$ is split with respect to some hyperplane $G(X_1, \ldots, X_d) = 0$. Then Algorithm 2 is applied on $\mathcal{Q}_1 := (\mathcal{Q} \wedge G(X_1, \ldots, X_d) \leq 0)$ and $\mathcal{Q}_2 := (\mathcal{Q} \wedge G(X_1, \ldots, X_d) \geq 0)$ separately. If there is no overflow and V is returned, the set of edges E is calculated and all the cycles are extracted. These cycles induce the periods, we are searching for. Note that the subsets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are again defined by finitely many non-strict inequalities so that they can be treated by Algorithm 1 in the same way as $\mathcal{Q}$.

In our setting we need to apply Algorithm 2 to the sets defined by the inequalities $\bar{T}_i$ ($i \in \{1, \ldots, 5\}$). All we need to specify is the subdividing strategy and the bound p for $|V|$. As for the subdividing strategy we

Algorithm 1 Calculation of the set of vertices of $G(X(\mathcal{Q}))$: VG

Require: $\mathcal{Q}$, p
Ensure: set of vertices V
1: $V \leftarrow \{\pm\mathbf{e}_j | j = 1, \ldots, d\}$
2: $M \leftarrow \emptyset$
3: **while** $V \neq M$ **do**
4: **if** $\#V > p$ **then**
5: **return**(Overflow)
6: stop calculation
7: **end if**
8: $N \leftarrow V \setminus M$
9: $M \leftarrow V$
10: **for all** $(x_1, \ldots, x_d) \in N$ **do**
11: $i \leftarrow \min_{(r_1,\ldots,r_d)\in X(\mathcal{Q})}(\lfloor -\sum_{k=1}^{d} x_k r_k \rfloor)$
12: $j \leftarrow \max_{(r_1,\ldots,r_d)\in X(\mathcal{Q})}(-\lfloor \sum_{k=1}^{d} x_k r_k \rfloor)$
13: $V \leftarrow V \cup \{(x_2, \ldots, x_d, k) | k \in \{i, \ldots, j\}\}$
14: **end for**
15: **end while**
16: **return**(V)

Algorithm 2 Search for all periods within an area $X(\mathcal{Q})$ (recursively): FP

Require: $\mathcal{Q}$
Ensure: Π list of cycles
1: $p \leftarrow$ suitable bound
2: $V \leftarrow \mathrm{VG}(\mathcal{Q}, p)$
3: **if** $\neg$(overflow) **then**
4: $E \leftarrow$ set of edges of $G(X(\mathcal{Q}))$
5: $\Pi \leftarrow$ periods induced by the cycles of $G(X(\mathcal{Q}))$
6: **else**
7: construct $\mathcal{Q}_1$, $\mathcal{Q}_2$
8: $\Pi \leftarrow \mathrm{FP}(\mathcal{Q}_1)$
9: $\Pi \leftarrow \Pi \cup \mathrm{FP}(\mathcal{Q}_2)$
10: **end if**
11: **return**(Π)

subdivide a given set in two parts as follows. Let

$$m_i := \min_{(x_1,x_2,x_3)\in X(\mathcal{Q})} x_i, i = 1, 2, 3,$$
$$M_i := \max_{(x_1,x_2,x_3)\in X(\mathcal{Q})} x_i, i = 1, 2, 3,$$

and $j \in \{1, 2, 3\}$ be the smallest index for which $M_j - m_j = \max(M_1 - m_1, M_2 - m_2, M_3 - m_3)$. The dividing hyperplane is now defined by

$$G(X_1, X_2, X_3) = 0 \text{ with } G(X_1, X_2, X_3) := X_j - \frac{M_j + m_j}{2}.$$

For the upper bound of the number of vertices it turns out that a choice depending on the quantities $M_j - m_j$ is convenient. In particular, we choose $p = \frac{20}{M_j - m_j}$. Then we get the following result

3.3.4 Lemma. $\mathrm{FP}(\bigwedge T_i)$ *terminates for each* $i \in \{1, \ldots, 5\}$.

Proof We implemented the algorithms for T_i with the above mentioned subdivision strategy and bounds in Mathematica®. The program is available on the authors' homepages [82]. □

3.3.5 Theorem. *$S_i \subset \mathcal{D}_3^0$ holds for all $i \in \{1, \dots, 5\}$.*

Proof For each $i \in \{1, \dots, 5\}$ we have that $X(\bigwedge \bar{T}_i)$ is a convex hull of finitely many points. Moreover, $X(\bigwedge \bar{T}_i) = \overline{S_i}$. Denote by Π_i the set of periods computed by the application of Algorithm 2 on $\bigwedge \bar{T}_i$. Hence Π_i includes all periods associated to polyhedra having non-empty intersection with $X(\bigwedge \bar{T}_i)$. Now, according to (6.0.5), each of these periods $\pi \in \Pi_i$ induces a system of inequalities $\mathcal{P}(\pi)$. It turns out that for each $\pi \in \Pi_i$ we have

$$X(\mathcal{P}(\pi) \wedge \bigwedge T_i) = \emptyset \text{ holds for each } i \in \{1, \dots, 5\}$$

(an easy way for checking this is to apply the cylindrical algebraic decomposition algorithm). Thus there is no period that yields a nonempty cutout intersecting with S_i and therefore $S_i \subset \mathcal{D}_3^0$. □

Covering the set $\mathcal{D}_3 \setminus \mathcal{D}_3^0$ by cutout polyhedra. Fix $Q_1, \dots, Q_{43}$ to be the sets of inequalities of the 43 polyhedra induced by the periods given in Table 3.1, where Q_j denotes just the reduced set of inequalities such that $X(\bigwedge Q_j)$ yields the corresponding polyhedron for any $j \in \{1, \dots, 43\}$. "Reduced" means that all the redundant inequalities are removed.

3.3.6 Remark. It is not really necessary to work with the reduced systems but the main algorithm works much faster and the reduction is not too difficult to realize.

Algorithm 3 Reducing a list of inequalities: RL

Require: P set of inequalities
Ensure: P reduced set of inequalities
1: **for all** inequalities $I \in P$ **do**
2: $P \leftarrow P \setminus I$
3: **if** $X(\bigwedge P \wedge \neg I) \neq \emptyset$ **then**
4: $P \leftarrow P \cup I$
5: **end if**
6: **end for**
7: **return**(P)

The algorithm simply uses the fact that an inequality I is redundant for a system $\mathcal{S} \wedge I$ if $X(\mathcal{S} \wedge I) = X(\mathcal{S})$ or, equivalently, $X(\mathcal{S} \wedge \neg I) = \emptyset$. Denote

the application of Algorithm 3 with parameter P by $\mathrm{RL}(P)$ (RL=**r**educe **l**ist of inequalities).

At the end of Section 3.2 we found a parametrization of $\overline{\mathcal{E}_3(1)}$. We saw that $\overline{\mathcal{E}_3(1)} = X(\bigwedge D)$ for

$$D := \{x + z \leq 1 + y, -1 - y \leq x + z, y - xz \leq 1 - x^2, z \leq 3, z \geq -3\}.$$

Let $\mathcal{P}$ be a list of sets of inequalities and G to be a set of inequalities. We want to verify if $\bigcup_{P\in\mathcal{P}} X(\bigwedge P)$ covers $X(\bigwedge G)$. This is equivalent to

$$X\left(\bigwedge G \wedge \neg \bigvee_{P\in\mathcal{P}} \bigwedge P\right) = \emptyset. \tag{3.3.2}$$

In principle we could do this verification directly. For computational reasons we are a little more restricted. (In fact the direct verification of (3.3.2) overcharges Mathematica®). A verification of a claim of the shape (3.3.2) can be done in a reasonable amount of time if $\#\mathcal{P} \leq 3$. We give an algorithm that solves this problem for general $\mathcal{P}$ and G by a subdivision process. The idea is to split the set $X(\bigwedge G)$ into suitable subsets and hope that each of these subsets is covered by a smaller number of sets. First we state Algorithm 4 which removes those sets from $\mathcal{P}$ that do not affect G, hence a set P is removed when $X(\bigwedge G) \cap X(\bigwedge P) = \emptyset$. Denote the application of this algorithm by $\mathrm{RS}(G, \mathcal{P})$ (RS=**r**emove inequalities with respect to a **s**et).

Algorithm 4 Removing those lists of inequalities that do not affect a given set G: RS

Require: G, $\mathcal{P}$
Ensure: $\mathcal{P}$ reduced list of inequalities
1: **for all** sets $P \in \mathcal{P}$ **do**
2: **if** $X(\bigwedge G \wedge \bigwedge P) = \emptyset$ **then**
3: $\mathcal{P} \leftarrow \mathcal{P} \setminus P$
4: **end if**
5: **end for**
6: **return**$(\mathcal{P})$

The main algorithm (Algorithm 5) is recursive. As an input we have again $\mathcal{P}$ and G of the usual shape, where $\mathcal{P}$ is reduced by Algorithm 4. Whenever the algorithm recognizes that a subset of $X(\bigwedge G)$ is not fully covered by the sets described in $\mathcal{P}$, it returns this subset. Denote the application by $\mathrm{VC}(G,\mathcal{P})$ (VC=**v**erify **c**overing). At first Algorithm 5 checks whether $\#\mathcal{P} \leq 3$. If this is the case we can verify whether (3.3.2) holds, otherwise we choose an arbitrary inequality $I \in \bigcup_{P\in\mathcal{P}} P$ such that $X(\bigwedge G \wedge I) \neq X(\bigwedge G)$. There are two possibilities:

- There is such an inequality I. Then $X(\bigwedge G)$ is split by adding I and $\neg I$, respectively, to G and Algorithm 5 is applied (recursively) on both of these subsets. Algorithm 4 is used to possibly reduce $\mathcal{P}$ for each of the subsets. These reduced sets form the second parameter.
- There is no such I. But this would mean that all the points of $X(\bigwedge G)$ suffice all inequalities of $\bigcup_{P\in\mathcal{P}} P$. This is equivalent to $X(\bigwedge G) \subset X(P)$ for any $P \in \mathcal{P}$ and this implies that G and $\mathcal{P}$ suffice the condition (3.3.2).

Now, whenever (3.3.2) is not fulfilled, the set $X(\bigwedge G)$ is not covered by $X(\bigvee_{P\in\mathcal{P}} \bigwedge P)$ and the algorithm returns the set $X(\bigwedge G)$. The application of Algorithm 5 terminates without any output if $X(\bigvee_{P\in\mathcal{P}} \bigwedge P)$ covers $X(\bigwedge G)$.

We can now state the main theorem of this subsection.

3.3.7 Theorem. *The algorithm* $\mathrm{VC}(D,\mathcal{P})$ *terminates without yielding any output for*

$$\mathcal{P} = \{Q_1, \ldots, Q_{43}, T_1, \ldots, T_5\}.$$

Proof We implemented the algorithms in Mathematica®. The program is available on the authors' homepages [82]. □

Theorem 3.3.7 shows all the periods together with our set to really cover all of $\overline{\mathcal{E}_3(1)}$ and thus cover $\mathcal{D}_3$. Thus, the cutout polyhedra $P(\pi_1), \ldots, P(\pi_{43})$ cover the whole set $\overline{\mathcal{E}_3(1)} \setminus \mathcal{S}$. Hence, in view of Theorem 3.3.5 we get that

$$\overline{\mathcal{E}_3(1)} \setminus \mathcal{S} \subset \bigcup_{1\leq i\leq 43} P(\pi_i).$$

Together with Theorem 3.3.5 this yields Theorem 8. and we are done.

Algorithm 5 Checks if a set is covered by the union of others (recursively): VC

Require: G, $\mathcal{P}$
Ensure: subsets of $X(\bigwedge G)$ that are not fully covered by $X(\bigvee_{P\in\mathcal{P}} \bigwedge P)$

if $\#\mathcal{P} \leq 3$ **then**
  if $X(G \wedge \neg \bigvee_{P\in\mathcal{P}} \bigwedge P) \neq \emptyset$ **then**
    return($X(\bigwedge G)$ is not fully covered)
  end if
else
  if $\exists I \in \bigcup_{P\in\mathcal{P}} P : X(\bigwedge G \wedge I) \neq \emptyset$ **then**
    $\mathrm{VC}(\mathrm{RL}(G \cap \{I\}), \mathrm{RS}(G \cap \{I\}, \mathcal{P})$
    $\mathrm{VC}(\mathrm{RL}(G \cap \{\neg I\}), \mathrm{RS}(G \cap \{\neg I\}, \mathcal{P})$
  end if
end if

Chapter 4

Cryptographic Protocol Building Blocks

This chapter details all the cryptographic primitives that are applied as building blocks of our election schemes presented in sections 5.3 and 5.4. Among these primitives one can find encryption schemes, ordinary and blind signature schemes and several zero-knowledge proofs. At the end of this chapter communication channels usually employed by voting schemes are described.

4.1 Encryption Schemes

The concept of public-key cryptography was invented by Whitfield Diffie and Martin Hellman in 1976 [24]. Since 1976, numerous public-key cryptography algorithms have been proposed.

4.1.1 Definition. A **cryptosystem** is a five-tuple $(\mathcal{PS}, \mathcal{CS}, \mathcal{KS}, \mathcal{EF}, \mathcal{DF})$, where the following conditions are satisfied:

- $\mathcal{PS}$ is a finite set of possible plaintexts
- $\mathcal{CS}$ is a finite set of possible ciphertexts
- $\mathcal{KS}$, the keyspace, is a finite set of possible keys

- $\mathcal{EF} = \{E_K | K \in \mathcal{KS}\}$ is a family of functions $E_K : \mathcal{PS} \mapsto \mathcal{CS}$. Its elements are called encryption functions.
- $\mathcal{DF} = \{D_K | K \in \mathcal{KS}\}$ is a family of functions $D_K : \mathcal{CS} \mapsto \mathcal{PS}$. Its elements are called decryption functions.
- For each $K \in \mathcal{KS}$, there is an encryption rule $E_K \in \mathcal{EF}$ and a corresponding decryption rule $D_K \in \mathcal{DF}$ such that $D_K(E_K(x)) = x$ for every plaintext $x \in \mathcal{PS}$.

4.1.1 RSA cryptosystem

RSA [71] is one of the most popular algorithm that works for encryption.

Let $N = P \cdot Q$, where P and Q are large primes. Let $\mathcal{PS} = \mathcal{CS} = \mathbb{Z}_N$, and define

$$\begin{aligned}\mathcal{KS} \quad = \quad & \{(N, P, Q, e, d) \mid N = P \cdot Q,\ (e, \phi(N)) = 1,\ e \cdot d \equiv 1 \pmod{\phi(N)}, \\ & 1 < e < \phi(N),\ 1 < d < \phi(N)\}.\end{aligned}$$

For $K = (N, P, Q, e, d)$, define

$$E_K(x) \equiv x^e \pmod{N}$$

and

$$D_K(y) \equiv y^d \pmod{N}$$

$(x, y \in \mathbb{Z}_N)$. The values N and e are public, and the values P, Q, d are secret.

The security of RSA depends wholly on the problem of factoring large numbers.

4.1.2 ElGamal cryptosystem

The ElGamal scheme [26] encryption scheme works in a finite cyclic group, where the discrete logarithm problem is computationally infeasible.

Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$. Let $\mathcal{PS} = G_Q$, $\mathcal{CS} = G_Q \times G_Q$, and define

$$\mathcal{KS} = \{(P, g, \alpha, h\ ) : h \equiv g^{\alpha} \pmod{P},\ h \in G_Q,\ \alpha \in \mathbb{Z}_Q\}.$$

The values P, g and h are public, and α is secret. For $K = (P, g, \alpha, h)$, and for a secret random number $k \in \mathbb{Z}_Q$, define

$$E_K(x, k) = (y_1, y_2),$$

where

$$y_1 \equiv g^k \pmod{P}$$

and

$$y_2 \equiv x \cdot h^k \pmod{P}.$$

For $y_1, y_2 \in G_Q$, define

$$D_K(y_1, y_2) = y_2 \cdot (y_1^{\alpha})^{-1} \pmod{P}.$$

ElGamal cryptosystem is non-deterministic, for the same plaintext there are several ciphertexts. The ciphertext depends on the plaintext and the random value k.

4.1.2 Definition. Let $\mathcal{PS}$ be the plaintext space and $\mathcal{CS}$ the ciphertext space such that $\mathcal{PS}$ is a group under the operation $\oplus$ and $\mathcal{CS}$ is a group under the operation $\otimes$. Let $E_r(m)$ denote encryption of the message m using random parameter r. An encryption scheme is $(\otimes, \oplus)$-*homomorphic*, if for given $E_{r_1}(m_1)$ and $E_{r_2}(m_2)$, there exists an r such that

$$E_{r_1}(m_1) \otimes E_{r_2}(m_2) = E_r(m_1 \oplus m_2).$$

Regarding ElGamal cryptosystem, the operation $\oplus$ is a multiplication modulo P and the operation $\otimes$ defined on cyphertexts is a multiplication modulo P per components.

$$E_{k_1}(m_1) \equiv (y_1, y_2),$$
$$E_{k_2}(m_2) \equiv (z_1, z_2),$$

where

$$y_1 \equiv g^{k_1} \pmod{P},$$
$$y_2 \equiv h^{k_1} \cdot m_1 \pmod{P},$$
$$z_1 \equiv g^{k_2} \pmod{P},$$
$$z_2 \equiv h^{k_2} \cdot m_2 \pmod{P},$$

and $E_{k_1}(m_1) \cdot E_{k_2}(m_2) = E_k(m_1 \cdot m_2)$, where $k = k_1 + k_2$.

Distributed ElGamal Key Generation

ElGamal cryptosystem can be easily modified into a non-threshold, distributed scheme [63] for n participants, that is the n participants together compute ElGamal public key without revealing the proper secure key or its shares to each other and they decrypt together the encrypted message. Underlying zero-knowledge proofs will be presented in section 4.4.

Input: P, Q, g, where P, Q are large primes such that $Q|P-1$ and G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$
Output: Public key: $h \pmod{P}$, public key shares $h_i \pmod{P}$, private key shares: $\alpha_i \pmod{Q}$

1. FOR $i = 1$ TO n DO
2. Each participant chooses $\alpha_i \in \mathbb{Z}_Q$ additive shares at random and computes $h_i \equiv g^{\alpha_i} \pmod{P}$ multiplicative shares
3. Each participant publishes $h_i \pmod{P}$ and zero-knowledge proof of knowing $\alpha_i \pmod{Q}$ on $\mathcal{BB}$
4. ENDFOR
5. $\alpha = \sum_{i=1}^{n} \alpha_i$ is the private key
6. $h \equiv \prod_{i=1}^{n} h_i \pmod{P}$ is the public key

Distributed ElGamal Decryption

Input: encrypted message: $(y_1 \pmod P, y_2 \pmod P)$, where $y_1 \equiv g^k \pmod P$ and $y_2 \equiv x \cdot h^k \pmod P$, public key shares: $h_i \pmod P$, private key shares: $\alpha_i \pmod Q$
Output: message: x

1. FOR $i = 1$ TO n DO
2. Each participant publishes $c_i \equiv y_1^{\alpha_i} \pmod P$ decryption share
3. Each participant publishes zero-knowledge proof of equality of h_i's and c_i's discrete logarithm
4. ENDFOR
5. $C \equiv \prod_{i=1}^{n} c_i \pmod P$
6. $x \equiv \frac{y_2}{C} \pmod P$

4.2 Signature Schemes

A signature scheme consists of three algorithm a key generation, a signing and a verification algorithm. During key generation the corresponding secret and public keys are defined, during the signing algorithm a message is signed by the secret key, and anybody with the knowledge of public key can run verification algorithm and decide whether the signature is valid or not.

4.2.1 Definition. A **signature scheme** is a five-tuple $(\mathcal{PS}, \mathcal{AS}, \mathcal{KS}, \mathcal{SF}, \mathcal{VF})$, where the following conditions are satisfied:

- $\mathcal{PS}$ is a finite set of possible messages.
- $\mathcal{AS}$ is a finite set of possible signatures.
- $\mathcal{KS}$, the keyspace, is a finite set of possible keys.
- $\mathcal{SF} = \{sig_K | K \in \mathcal{KS}\}$ is a family of functions $sig_K : \mathcal{PS} \mapsto \mathcal{AS}$. Its elements are called signing algorithms.

- $\mathcal{VF} = \{ver_K | K \in \mathcal{KS}\}$ is a family of functions $ver_K : \mathcal{PS} \times \mathcal{AS} \mapsto \{true, false\}$. Its elements are called verification algorithms.
- For each $K \in \mathcal{KS}$, there is a signing algorithm $sig_K \in \mathcal{SF}$ and a corresponding verification algorithm $ver_K \in \mathcal{VF}$ such that the following equation is satisfied for every message $x \in \mathcal{PS}$ and for every signature $y \in \mathcal{AS}$:

$$ver(x, y) = \left\{ \begin{array}{ll} true, & \text{if } y = sig(x); \\ false, & \text{if } y \neq sig(x). \end{array} \right\}.$$

4.2.1 RSA signature scheme

RSA public-key cryptosystem can be used to provide digital signatures [71].

Let $N = P \cdot Q$, where P and Q are large primes. Let $\mathcal{PS} = \mathcal{AS} = \mathbb{Z}_N$, $M : \mathbb{Z}_N \mapsto \mathbb{Z}_N$ a hash function, and define

$$\begin{aligned} \mathcal{KS} \;=\; & \{(N, P, Q, e, d) \mid N = P \cdot Q, \text{ where } P, Q \text{ prime}, \ (e, \phi(N)) = 1, \\ & e \cdot d \equiv 1 \pmod{\phi(N)}, \ 1 < e < \phi(N), \ 1 < d < \phi(N)\}. \end{aligned}$$

The values N and e are public, and the values P, Q, d are secret. For $K = (N, P, Q, e, d)$, and $x \in \mathbb{Z}_N$ message define

$$sig_K(x) \equiv M(x)^d \pmod{N}$$

and

$$ver_K(x, y) = true \Leftrightarrow M(x) \equiv y^e \pmod{N}$$

$(x, y \in \mathbb{Z}_N)$.

4.2.2 ElGamal Signature Scheme

We now describe ElGamal Signature scheme [26].

Let P and Q large primes such that $Q|P - 1$ and let denote G_Q as $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and choose g an arbitrary element such that $g \in G_Q$. Let $\mathcal{PS} = G_Q$, $\mathcal{AS} = G_Q \times \mathbb{Z}_Q$, and define

$$\mathcal{KS} = \{(P, Q, g, \alpha, h\ ) : h \equiv g^\alpha \pmod{P}, \ h \in G_Q, \ \alpha \in \mathbb{Z}_Q\}.$$

The values P, Q, g and h are public, and α is secret. For $K = (P, Q, g, \alpha, h)$, $x \in G_Q$ message, and for a secret random number $k \in \mathbb{Z}_Q$, define

$$sig_K(x, k) = (\gamma, \delta),$$

where

$$\gamma \equiv g^k \pmod{P}$$

and

$$\delta = (x - \alpha \cdot \gamma) \cdot k^{-1} \pmod{Q}.$$

For $x, \gamma \in G_Q$ and $\delta \in \mathbb{Z}_Q$, define

$$ver_K(x, \gamma, \delta) = true \Leftrightarrow h^\gamma \cdot \gamma^\delta \equiv g^x \pmod{P}.$$

Several digital signature schemes are developed based on discrete logarithm problem. Along with thousands of other signature schemes, they are part of the same family. In [37] all these approaches are integrated in a Meta-ElGamal signature scheme. One of the generalization is as follows.

Generalization of ElGamal Signature Scheme
Let P and Q large primes such that $Q|P-1$ and let denote G_Q as $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and choose g an arbitrary element such that $g \in G_Q$. Let $\mathcal{PS} = G_Q$, $\mathcal{AS} = G_Q \times \mathbb{Z}_Q$, and define

$$\mathcal{KS} = \{(P, Q, g, \alpha, h\ ) : h \equiv g^\alpha \pmod{P},\ h \in G_Q,\ \alpha \in \mathbb{Z}_Q\}.$$

The values P, Q, g and h are public, and α is secret. For $K = (P, Q, g, \alpha, h)$, $x \in G_Q$ message, and for a secret random number $k \in \mathbb{Z}_Q$, define

$$sig_K(x, k) = (\gamma, \delta),$$

where

$$\gamma \equiv g^k \pmod{P}$$

and δ derives from the following congruence:

$$A \equiv \alpha \cdot B + k \cdot C \pmod{Q},$$

where we choose A, B, C as a permutation of the parameters (x, γ, δ).
For $x, \gamma \in G_Q$ and $\delta \in \mathbb{Z}_Q$, define

$$ver_K(x, \gamma, \delta) = true \Leftrightarrow h^B \cdot \gamma^C \equiv g^A \pmod{P},$$

By this method we get six possible signatures schemes, denoted by $No\ 1, 2, 3, 4, 5, 6$. The following table lists all variants and the corresponding signature and verification congruences.

No	A	B	C	signature	verification
1	x	γ	δ	$x \equiv \alpha \cdot \gamma + k \cdot \delta \pmod{Q}$	$g^x \equiv h^\gamma \cdot \gamma^\delta \pmod{P}$
2	x	δ	γ	$x \equiv \alpha \cdot \delta + k \cdot \gamma \pmod{Q}$	$g^x \equiv h^\delta \cdot \gamma^\gamma \pmod{P}$
3	δ	γ	x	$\delta \equiv \alpha \cdot \gamma + k \cdot x \pmod{Q}$	$g^\delta \equiv h^\gamma \cdot \gamma^x \pmod{P}$
4	δ	x	γ	$\delta \equiv \alpha \cdot x + k \cdot \delta \pmod{Q}$	$g^\delta \equiv h^x \cdot \gamma^\gamma \pmod{P}$
5	γ	δ	x	$\gamma \equiv \alpha \cdot \delta + k \cdot \gamma \pmod{Q}$	$g^\gamma \equiv h^\delta \cdot \gamma^x \pmod{P}$
6	γ	x	δ	$\gamma \equiv \alpha \cdot x + k \cdot \delta \pmod{Q}$	$g^\gamma \equiv h^x \cdot \gamma^\delta \pmod{P}$

Table 1: Generalized ElGamal Signature Scheme

4.2.3 Schnorr Signature Scheme

Schnorr signature scheme [80] gets its security also from the difficulty of calculating discrete logarithms.

Let P be a 512-bit prime such that the discrete logarithm problem in $\mathbb{Z}_P$ is intractable, and let Q be a 160-bit prime divides $P-1$. Let denote G_Q as $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and choose g an arbitrary element such that $g \in G_Q$. Let $M : G_Q \times G_Q \mapsto \mathbb{Z}_Q$ hash function. Let $\mathcal{PS} = G_Q$, $\mathcal{AS} = G_Q \times \mathbb{Z}_Q$, and define

$$\mathcal{KS} = \{(P, Q, g, \alpha, h\ ) : h \equiv g^\alpha \pmod{P},\ h \in G_Q,\ \alpha \in \mathbb{Z}_Q\}.$$

The values P, Q, g and h are public, and α is secret. For $K = (P, Q, g, \alpha, h)$, $x \in G_Q$ message, and for a secret random number $k \in \mathbb{Z}_Q$, define

$$\begin{aligned} sig_K(x, k) &= (\gamma, \delta), \\ \gamma &\equiv g^k \pmod{P}, \\ \delta &\equiv k - \alpha \cdot c \pmod{Q}, \end{aligned}$$

where $c = M(x, \gamma)$. For $x, \gamma \in G_Q$ and $\delta \in \mathbb{Z}_Q$ and

$$ver_K(x, \gamma, \delta) = true \Leftrightarrow \gamma \equiv g^\delta \cdot h^c \pmod{P},$$

The signing algorithm of Schnorr signature scheme can be transformed into a three round interactive protocol between the signer and a user in the following way:

1. The signer picks $k \in \mathbb{Z}_Q$ randomly, and sends $\gamma \equiv g^k \pmod{P}$ along with the message $x \in G_Q$ to the user.
2. The user calculates $c = M(x, \gamma)$ challenge and transmits it to the signer.
3. The signer replies $\delta \equiv k - \alpha \cdot c \pmod{Q}$.

4.3 Blind Signature Schemes

The concept of blind signature schemes was introduced by Chaum in 1982 [19]. These schemes can be applied in payment systems and for electronic voting schemes. In a blind signature scheme an owner wishes to get a digital signature on his message from a notary, but the notary does not have information about the message itself. After receiving the signed message the notary will not find relationship between the blinded and unblinded signature. Usually the key-generation, the signature generation and the verification algorithms are the same as in the corresponding non-blind signature schemes, but the blind signing algorithm is always an interactive protocol between the user and the signer.

4.3.1 RSA Blind Signature Scheme

This signature scheme is presented in [68].
Let $N = P \cdot Q$, where P and Q are large primes. Let $\mathcal{PS} = \mathcal{AS} = \mathbb{Z}_N$, $M : \mathbb{Z}_N \mapsto \mathbb{Z}_N$ a hash function, and define

$$\begin{aligned}\mathcal{KS} \;=\; & \{(N, P, Q, e, d) \mid N = P \cdot Q,\ (e, \phi(N)) = 1,\ e \cdot d \equiv 1 \pmod{\phi(N)},\\ & 1 < e < \phi(N),\ 1 < d < \phi(N)\},\end{aligned}$$

The values N and e are public, and the values P, Q, d are secret. For $K = (N, P, Q, e, d)$, $x \in \mathbb{Z}_N$ message,

1. the user chooses a random value $k \in \mathbb{Z}$, and sends $\tilde{x} \equiv k^e \cdot M(x) \pmod{N}$ to the signer.

2. The signer replies with $sig_K(\tilde{x}) \equiv (\tilde{x})^d \pmod{N}$,

3. the user generates signature $sig_K(x) \equiv sig_K(\tilde{x}) \cdot k^{-1} \pmod{N}$.

The verification algorithm is $ver_K(x, y) = true \Leftrightarrow M(x) \equiv y^e \pmod{N}$ $(x, y \in \mathbb{Z}_N)$.

4.3.2 ElGamal Blind Signature Scheme

Let us consider the generalized ElGamal signature scheme. The generalized blind signature scheme in [37] is as follows.

Let P and Q large primes such that $Q|P-1$ and let denote G_Q as $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and choose g an arbitrary element such that $g \in G_Q$. Let $\mathcal{PS} = G_Q$, $\mathcal{AS} = G_Q \times \mathbb{Z}_Q$, and define

$$\mathcal{KS} = \{(P, Q, g, \alpha, h\ ) : h \equiv g^\alpha \pmod{P},\ h \in G_Q,\ \alpha \in \mathbb{Z}_Q\}.$$

The values P, Q, g and h are public, and α is secret. For $K = (P, Q, g, \alpha, h)$, $x \in G_Q$ message, $sig_K(x) = (\gamma, \delta)$ signature is generated in the following way:

1. The notary chooses a random number $\tilde{k} \in \mathbb{Z}_Q$ and computes $\tilde{\gamma} \equiv g^{\tilde{k}} \pmod{P}$ and sends it to the user.

2. The user chooses $a, b \in \mathbb{Z}_Q$ random numbers and calculates $\gamma \equiv \tilde{\gamma}^a \cdot g^b \pmod{P}$.

3. The user computes $\tilde{x} \equiv \psi(a, b, x, \gamma, \tilde{\gamma})$ and transfers it to the notary.

4. The notary signs $\tilde{x}$ with congruence of $\tilde{A} \equiv \alpha \cdot \tilde{B} + \tilde{k} \cdot \tilde{C} \pmod{Q}$, where $\tilde{A}, \tilde{B}, \tilde{C}$ is a permutation of values of $\tilde{\delta}, \tilde{x}, \tilde{\gamma}$ and replies $\tilde{\delta}$ back.

5. The user retrieves $\delta \equiv \theta(a, b, x, \gamma, \tilde{x}, \tilde{\gamma}, \tilde{\delta})$.

The verification congruence of the signature is $\alpha^A \equiv h^B \cdot \gamma^C \pmod{P}$, where A, B, C is a permutation of x, δ, γ.

Table 2 shows the method of calculating ψ and θ functions for the possible cases of *Table 1*. Note that considering *Table 1* for those cases, where δ appears in C we cannot get blind signature schemes, because δ and $\tilde{\delta}$ are not allowed as arguments of function ψ.

No	ψ	θ
2	$x \equiv a \cdot \tilde{\gamma}^{-1} \cdot \tilde{x} \cdot \gamma + b \cdot \gamma \pmod{Q}$	$\delta \equiv a \cdot \tilde{\gamma}^{-1} \cdot \tilde{\delta} \cdot \gamma \pmod{Q}$
3	$\gamma \equiv a \cdot \tilde{x}^{-1} \cdot \tilde{\gamma} \cdot x \pmod{Q}$	$\delta \equiv a \cdot \tilde{x}^{-1} \cdot \tilde{\delta} \cdot x + b \cdot x \pmod{Q}$
4	$x \equiv a \cdot \tilde{\gamma}^{-1} \cdot \tilde{x} \cdot \gamma \pmod{Q}$	$\delta \equiv a \cdot \tilde{\gamma}^{-1} \cdot \tilde{\delta} \cdot \gamma + b \cdot \gamma \pmod{Q}$
5	$\gamma \equiv a \cdot \tilde{x}^{-1} \cdot \tilde{\gamma} \cdot x + b \cdot x \pmod{Q}$	$\delta \equiv a \cdot \tilde{x}^{-1} \cdot \tilde{\delta} \cdot x \pmod{Q}$

Table 2: Generalized ElGamal Blind Signature Scheme

4.3.3 Schnorr Blind Signature Scheme

This signature scheme is described in [68].
Let P, Q two large prime number, such that $Q|P-1$. They are published together an element $g \in \mathbb{Z}_P^*$ of order Q. Let denote G_Q the cyclic group generated by g and $M : G_Q \times G_Q \mapsto \mathbb{Z}_Q$ a hash function. Let $\mathcal{PS} = G_Q$, $\mathcal{AS} = G_Q \times \mathbb{Z}_Q$, and define

$$\mathcal{KS} = \{(P, Q, g, \alpha, h\ ) : h \equiv g^{\alpha} \pmod{P},\ h \in G_Q,\ \alpha \in \mathbb{Z}_Q\}.$$

The values P, Q, g and h are public, and α is secret. For $K = (P, Q, g, \alpha, h)$, $x \in G_Q$ message, $sig_K(x) = (c, \delta, \gamma)$ signature is generated in the following way:

1. The signer chooses a random number $\tilde{k} \in \mathbb{Z}_Q$ and sends commitment $\tilde{\gamma} \equiv g^{\tilde{k}} \pmod{P}$ to the user.

2. The user chooses random elements $a, b \in \mathbb{Z}_Q$ and $\gamma \equiv \tilde{\gamma} \cdot g^a \cdot h^{-b} \pmod{P}$ and computes $c = M(x, \gamma)$ and sends challenge $\tilde{c} \equiv c + b \pmod{Q}$ to the signer.

3. The signer returns value $\tilde{\delta}$, such that $\tilde{\delta} \equiv \tilde{k} - \tilde{c} \cdot \alpha \pmod{Q}$.

4. The user calculates $\delta \equiv \tilde{\delta} - a \pmod{Q}$ and $c \equiv \tilde{c} - b \pmod{Q}$ and outputs (c, δ, γ).

The verification algorithm is

$$ver_K(c, \delta, \gamma) = true \Leftrightarrow \gamma \equiv g^{\delta} \cdot h^{c} \pmod{P}.$$

4.4 Zero-knowledge Proofs

Let P and Q large primes such that $Q|P-1$ and let denote G_Q as $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and choose g an arbitrary element such that $g \in G_Q$.

4.4.1 Proof of knowledge of a discrete logarithm

Let $h \equiv g^{\alpha} \pmod{P}$ and h, g are public parameters and α is known only by a prover $\mathcal{A}$. $\mathcal{A}$ wants to show to a verifier $\mathcal{B}$ that he knows discrete logarithm α. The classic protocol in [80] is the following:

1. $\mathcal{A}$ chooses a random number $k \in \mathbb{Z}_Q$ and sends $a \equiv g^k \pmod{P}$ to $\mathcal{B}$.

2. $\mathcal{B}$ chooses a challenge $c \in \mathbb{Z}_Q$ at random and transmits it to $\mathcal{A}$.

3. $\mathcal{A}$ sends $l \equiv k + c \cdot \alpha \pmod{Q}$ to $\mathcal{B}$.

4. $\mathcal{B}$ verifies whether $g^l \equiv a \cdot h^c \pmod{P}$.

4.4.2 Proof of equality of two discrete logarithm

A participant A wants to prove a possession of a common discrete logarithm $\alpha \in \mathbb{Z}_Q$ satisfying $a \equiv g^{\alpha} \pmod{P}$ and $b \equiv h^{\alpha} \pmod{P}$. An efficient protocol is described in [20].

1. A chooses $k \in \mathbb{Z}_Q$ at random and sends $u \equiv g^k \pmod{P}$ and $v \equiv h^k \pmod{P}$ to verifier B.

2. B chooses a challenge $c \in \mathbb{Z}_Q$ randomly and transmits it to A.
3. A sends $l \equiv k + c \cdot \alpha \pmod{Q}$ to B.
4. B checks whether $g^l \equiv u \cdot a^c \pmod{P}$ and $h^l \equiv v \cdot b^c \pmod{P}$.

4.4.3 Proof of encrypted value is 1 out of n values

User $\mathcal{A}$ wants to give a proof that an encrypted value $(G, H \cdot C_i)$, where $G \equiv g^\alpha \pmod{P}$ and $H \equiv h^\alpha \cdot C_i \pmod{P}$ is an ElGamal encryption of the ith element of the given $C_1, \cdot, C_n$ values. This zero-knowledge proof is described in [52].

1. Prover A chooses $w \in \mathbb{Z}_Q$ at random and calculates $a_i \equiv g^w \pmod{P}$ and $b_i \equiv h^w \pmod{P}$.
2. FOR $j = 1$ TO n DO
3. IF $j \neq i$ DO
4. A chooses $d_j, r_j \in \mathbb{Z}_Q$ at random and calculates $a_j \equiv g^{r_j} \cdot G^{d_j} \pmod{P}$ and $b_j \equiv h^{r_j} \cdot (\frac{H \cdot C_i}{C_j})^{d_j} \pmod{P}$.
5. ENDIF
6. ENDFOR
7. A computes $(\mathsf{A}, \mathsf{B}) = (a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n)$ and sends $(\mathsf{A}, \mathsf{B}), (G, H \cdot C_i)$ to the verifier $\mathcal{B}$.
8. $\mathcal{B}$ chooses random challenge $c \in \mathbb{Z}_Q$ and sends it back to A.
9. A computes $d_i \equiv c - \sum_{j=1, j \neq i}^{n} d_j \pmod{Q}$ and $r_i \equiv w - \alpha \cdot d_i \pmod{Q}$ and sends $(\mathsf{D}, \mathsf{R}) = (d_1, r_1), (d_2, r_2), \cdots, (d_n, r_n)$ to B.
10. B verifies whether $c \equiv \sum_{j=1}^{n} d_j \pmod{Q}$.
11. B checks whether
    FOR $j = 1$ TO n DO
    $a_j \equiv g^{r_j} \cdot G^{d_j} \pmod{P}$ and
    $b_j \equiv h^{r_j} \cdot (\frac{H \cdot C_i}{C_j})^{d_j} \pmod{P}$
    ENDFOR

This zero-knowledge proof can be modified into non-interactive with

$$c = M(a_1||\cdots||a_n||b_1||\cdots||b_n||G||H \cdot C_i||g||h),$$

where $M()$ is a one-way hash function.

4.5 Communication Channels

During a cryptographic protocol parties involved exchange information through various types of communication channels. Electronic voting schemes besides the well-known public, and secure channels apply special ones like untappable, anonymous channels, voting booths and bulletin boards.

Public channel. Public channels transmit all information without applying cryptographic algorithm. Attackers are able to tap the message, and the identity of the sender can be traced back. In election schemes all the messages to the bulletin board are sent through public channels.

Secure channel. In order to realize a secure channel between two parties, first the participants run a key-exchange protocol to obtain a session key. The sender encrypts the message and concatenates it with a tag computed by applying a message authentication function to the ciphertext. Encryption and authentication are done via keys derived from the session key. Verification and decryption are done analogously.

Bulletin board. Bulletin board ($\mathcal{BB}$) is publicly readable. Voters, authorities can write into their section and nobody can modify the content of it.

Anonymous channel. This channel guarantees the anonymity of the sender. Receiver of the message that has been sent through an anonymous channel does not have any information about the identity of the sender. Especially, anonymous return channels allow two parties even to have a complete conversation, the receiver may reply to the sender. Realization of this channel is described in [32] based on a mix-net approach.

Untappable channel. This channel is a one-way physical apparatus providing perfect secrecy in an information-theoretic sense. Realizing an untappable channel in practice is considerably problematic. It might be achieved either by being physically untappable or by implementing

information-theoretic encryption. One well-known realization of perfect secrecy is the Vernam One-time Pad. The major problem is the key must be random, equal to the length of the message and can never be used again. Besides the problem of key distribution and storage, the perfect synchronization of the sender and receiver should be assured.

Voting booth. Voting-booths are the two-way version of the untappable channels. Besides supplying perfect secrecy they allow a voter interactively communicate with an authority.

Public, secure, anonymous channels and bulletin boards can be implemented in practice. Several authors in the literature have pointed out the difficulty of the implementation of untappable channels and voting booths [54].

Chapter 5

Cryptographically Secure Electronic Elections

This chapter starts with characterization of electronic voting requirements, then specification of the participants can be found. The last two sections deal with our main results, a coercion-resistant voting scheme based on blind signatures and a receipt-free homomorphic election scheme.

The results of this chapter are based on [38] and [39].

There are several election protocols using blind signatures that possess all basic requirements including verifiability, eligibility, unreusability, privacy etc., but not receipt-freeness. ([28],[58]) Most of the receipt-free schemes in literature make some basic assumptions about the communication channel between the voter and the election authorities. They apply untappable channels or voting booths ([59]). These communication channels are not practical. The other solution in order to achieve receipt-freeness is to employ tamper-resistant hardware ([54]). Our protocol in 5.3 does not require untappable channels or voting booths, voters use anonymous channels [61] that can be realized in practice using mix-nets. It does not rely on tamper-resistant hardware either and it does not apply any complex cryptographic primitives like zero-knowledge protocols, secret sharing or threshold cryptosystems like [21],[22],[41],[42] etc. do. It satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability and coercion-resistance.

The scheme in 5.4 is a homomorphic encryption model based on [22] that is not possessing the property of receipt-freeness or uncoercibility. Lee and Kim in [52] gave a solution for receipt-freeness applying an honest verifier. Hirt and Sako in [36] use an untappable channel to achieve it. This scheme does not employ voting booths or untappable channels, it requires an anonymous return channel [32], which is based on a mix-net approach, hence it can be implemented in practice. This channel has acceptable performance, four times the computational cost of a basic re-encryption mix-net. We do not suppose the existence of an honest verifier, either. Our scheme satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability, receipt-freeness, uncoercibility and protects against randomization and forced-abstention attacks.

Electronic voting schemes usually consist of three main stages: Authorizing, Voting and Tallying stages. During the *Authorizing stage* all system parameters, secret and public keys are generated and the voter roll is created with the list candidates. In the *Voting stage* the voter forms his ballot containing the vote and sends it to a voting authority through the channel he can use. After the deadline, during the *Tallying stage* authorities use their public and secret information and count the votes and publish the result.

In traditional elections, a voting booth not only allow voters to keep their vote secret, but it prevents vote-buying and coercion. The notions of *receipt-freeness* and *uncoercibility* were introduced by Benaloh and Tuinstra [12]. The property of receipt-freeness ensures that an attacker is not able to trace back voter's exact behavior, therefore a vote-buyer (coercer) does not obtain a reasonable proof. Hirt and Sako [36] showed that [12] does not possess receipt-freeness and introduced a receipt-free voting based on homomorphic encryption. Okamoto [58] proposed a voting scheme which he himself later showed to lack the postulated receipt-freeness, a repaired version using blind signatures appears in [59]. Lee and Kim [52] proposed a receipt-free version of [22] keeping optimal performance, privacy, robustness and universal verifiability. Sako and Kilien [74] proposed a multi-authority receipt-free scheme applying a mix network and also homomorphic encryption for tallying. Mix-net is used for tallying in [42] and at some point during the voting process voters post ballot to the bulletin board via anonymous channel. In [42] property of *coercion-resistance* is introduced. A coercion-resistant scheme gives a possibility for the voter to cheat an adversary who instructs him to vote in a given manner, but the

adversary cannot determine whether the voter behaved as instructed, even if the adversary asks the voter to divulge his private keying material or to abstain from voting.

5.1 Requirements

In order to be functional in practice, an electronic voting scheme has to satisfy not only all the standard features of the conventional paper-based voting methods, but also should provide more efficient voting services. E-voting comparing to the traditional elections allows adversaries to intrude the voting process in an easier way, even if there is a small security gap in the design. Thus the scheme should be protected against these techniques, the requirements are as follows:

Eligibility. Only eligible voters are allowed to cast votes.

Privacy. All votes remain secret, no one is able to link a vote to the voter, who has casted it. No considerably large coalition of participants not containing the voter himself can gain any information about a voter's vote.

Unreusability. Every eligible voter can cast at most one vote. No one can vote for anyone else.

Fairness. No participants can gain any knowledge about the partial tally during the voting stage, since knowledge of any intermediate result about the election can influence the voters.

Robustness. No participant can disrupt the election. Once a voter cast a vote, no alternation to this vote is permitted. Moreover all valid votes will be counted, whereas all invalid ones will be detected and not counted in the final tally.

Individual verifiability. Each eligible voter is able to verify that his vote was committed as intended and made into the final tally as cast.

Universal verifiability. Any participant or passive observer can check that the election is fair, the final result is exactly the sum of the valid votes.

Receipt-freeness, Uncoercibility. Before the election an adversary may bribe the voter with a demand of casting his favorite vote. This scenario is called vote-buying. Receipt-freeness avoids vote-buying. An adversary can also force the voter to cast a particular vote by threatening him. Uncoercibility means coercers cannot menace voters. These requirements

should be achieved in a way, that during the election a coercer can observe all public information and communication between the voter and the authorities and can even order the voter how he should behave during the voting process, even supplying him the random bits.

The exact definition of receipt-freeness is quoted from [59]:

Given published information $\mathcal{X}$ (public parameters and information on the bulletin board), adversary $\mathcal{C}$ interactively communicates with a voter V in order to force V to cast $\mathcal{C}$'s favorite vote c^* to an authority $\mathcal{A}$, and finally $\mathcal{C}$ decides whether to accept $View(\mathcal{X} : V)$ or not, and $\mathcal{A}$ decides whether to accept c^* or not. The coercer gets any message from the bulletin board immediately after it is put on the board. $View(\mathcal{X} : V)$ means published information $\mathcal{X}$, c^* and messages that $\mathcal{C}$ receives and sends communicating with V including random bits employed during the voting process.

5.1.1 Definition. A voting system is receipt-free, if there exists a voter V, such that for any adversary $\mathcal{C}$, voter V can cast c $(c \neq c^*)$ which is accepted by the authority $\mathcal{A}$ under the condition that $View(\mathcal{X} : V)$ is accepted by $\mathcal{C}$.

We suppose that a coercer knows public parameters appearing on the bulletin board, vote c^*, random bits predefined by him and encrypted messages sent by the voter on public channels. Receipt-freeness means $View(\mathcal{X} : V)$ should be prepared in a way that, if a coercer makes all calculations with all the data that he possesses, then no inaccurate count should turn up. A coercer is not able to monitor each communication channel being used during the voting process, hence encrypted data sent through an anonymous channel is not revealed to him. At the same time a ballot is accepted by an authority, if it has confirmed all necessary information and validity of ballots.

There are several real-word attacks in [42] enumerated below:

Randomization attack. An attacker coerces a voter to submit randomly formed ballot. In this attack it is not possible to learn what candidate the voter casts a ballot for. The effect of this attack is to cancel the voter's vote with large probability.

Forced-abstention attack. An attacker forces a voter to abstain from voting. This attack happens if an adversary is able to follow who is eligible for voting and who has already voted. Being aware of this knowledge he threatens voters and effectively excludes them from the voting process.

Simulation attack. In this attack an adversary coerces or bribes the voter to reveal his private keying material and then pretends to be the voter and casts his own favorite vote.

5.1.2 Definition. A scheme is called coercion-resistant if it offers not only receipt-freeness, but also defense against randomization, forced-abstention and simulation attacks.

5.2 Participants

Several participants contribute in an election system. If we had one absolutely reliable authority we would need no more authorities, and the voting process would be very simple. Since the situation is different in practice, the responsibility should be shared among several authorities. It is crucial to consider how much a participant can be trusted.

Voters. Let denote voters by $\mathcal{V} = \{V_1, V_2, \ldots, V_m\}$. A voter wants to have the guarantee that his vote is counted in the final tally and if a fraud is suspected there should be a possibility to make his claim. Authorities do not trust voters at all. We assume that the voter is not observed while casting his vote. Attacks, where a coercer is present or the voter is being recorded by a camera (e.g. cell phone camera) in the moment of voting is not considered.

Registry. Registry denoted by $\mathcal{R}$ is responsible for managing the authorizing stage. It checks voters' eligibility in person, supervising private and public key-generation for voting authorities participating in the election. Besides Registry supervises key-generation, reveals public keys to participants, also sets the necessary parameters for the whole election.

Voting Authorities. Several authorities denoted by $\mathcal{A} = \{A_1, A_2, \ldots, A_s\}$ are involved in carrying out the voting process. They have large computing power and possibility of storing large amount of data. Employees of the authorities may act as voters, too.

Adversaries. Any participant or group of them might be malicious and try to distract the elections or to achieve a favorable voting result even in an illegal way. Voters or even members of the voting authorities may become attackers. An attacker can also be an observer who would threaten or even pay participants to vote in a way he demands it.

5.3 A Coercion-Resistant Voting Scheme Based on Blind Signatures

In the following election scheme besides voters a Voting Authority and a Registry participate. The Registry manages the Authorizing stage and the Tallying stage, too.

Assumptions

1. The security of the proposed scheme relies on the correctly generated public and secret key pairs for the voters (PK_V, SK_V) and for the Voting Authority (PK_A, SK_A), too. It is also assumed that the Registry gives private key information only to the proper participants.

2. Since the responsibility of the security is shared, we suppose that the Registry and the Voting Authority do not collude. They both follow the steps of the protocol, not providing more information to each other that they are supposed to.

3. An adversary may coerce a voter to cast his vote in a prescribed manner. He can request voter's credential (V_{ID}, SK_V) right after the registration phase and dictates all random parameters (x, a) for the voter.

4. We suppose that voters 'personally' participate in the election. The adversary may not continuously watch over the shoulder of the voter, monitor his hard-drive, etc. During the voting there is a moment when the voter is alone and not being watched. A coercer is able to communicate with a voter right after the registration phase, and before and after the election.

5. The Voting Authority is honest in a sense that it does not collaborate with an adversary, does not give any information about the election and it does not generate spurious votes.

5.3.1 Protocol description

The proposed election procedure consists of three distinctive stages: *Authorizing*, *Voting* and *Tallying*.

During the Authorizing stage the voter authenticates himself and receives his credentials, the Voting Authority gets the voter roll containing the corresponding public keys and all system parameters are generated.

During the Voting stage voters create their ballots. Voting Authority checks eligibility of the voters and if they have already voted before. Voters receive their encrypted ballots signed by the Voting Authority, if a fraud is detected the voter makes a claim. At the end voters pass the corresponding decrypting keys of the encrypted ballots to the Registry. Ballots and bulletin board information are passed through an anonymous channel.

During the Tallying stage the Voting Authority sends encrypted ballots to the Registry. The ballots are being decrypted and the final results with the votes are listed on the bulletin board. Voters confirm that their ballots are on the bulletin board. If his ballot is not listed correctly, he makes a claim.

During the voting process public and anonymous channels are used and encrypted messages are sent. For the communication between the voters and the Voting Authority instead of higher degree residue encryption the more efficient discrete logarithm encryption is recommended. Let denote an encryption with public key PK by E_{PK}.

Let define a candidate slate to be an ordered set of n distinct identifiers $\{C_1, C_2, \ldots, C_n\}$, each of which corresponds to a voter choice, typically a candidate or party name.

Functions

Several functions are applied in the proposed election scheme. Let denote P, Q large primes, where $Q|(P-1)$ and $g \in \mathbb{Z}_P^*$ of order Q. The details of these functions are as follows:

Voting. Function $vote(V_{ID}, SK_V, x, a, C_i) \rightarrowtail ballot$ takes the voter's identification number V_{ID}, secret key SK_V, two randomly chosen parameter x, a and C_i as input and outputs the ballot. The form of the ballot is $(V_{ID}||r||y, V_{ID}||v)$, where

$$
\begin{aligned}
r &= E_{SK_V}(g) \\
y &\equiv g^{-x} \pmod{P} \\
v &\equiv y^a \cdot C_i \pmod{P}
\end{aligned}
$$

and || is the notation of concatenation. This function generates the ballot itself being processed by the Voting Authority.

Eligibility. Function $ifeligible(PK_V, r) \rightarrowtail \{0,1\}$ takes the voter's public key PK_V and the received element r as input. It returns 1 if

$$D_{PK_V}(r) = g$$

and 0 if the congruence above is not satisfied. This function checks if a voter is eligible for voting or not, *i.e.* if he possesses the proper private keying material V_{ID}, SK_V.

Verification. The function $verify(PK_V, z, s, y) \rightarrowtail \{0,1\}$ calculates if

$$PK_V^z \equiv g^s \cdot y \qquad (\text{mod } P)$$

congruence holds. It outputs 1 if it is correct and 0 otherwise. This function verifies if s sent by the voter is calculated well and by the same voter who previously *voted* with value y and public key PK_V, where element z is randomly generated by the Voting Authority.

In the following we discuss each step in more details. Figure 5.1 shows the steps of the voting protocol.

Authorizing stage

$$\mathcal{R} \longrightarrow \mathcal{V} : (V_{ID}, SK_V, PK_A)$$
$$\mathcal{R} \longrightarrow \mathcal{A} : (V_{ID}, PK_V)$$

Before the voting process the voter must register with Registry verifying his identity. Registry issues a credential to each eligible voter and prepares a list of registered voters. A credential consists of voter's ElGamal secret key SK_V, an identification number V_{ID}, public key of the Voting Authority PK_A. The voter roll contains key pairs (V_{ID}, PK_V), where PK_V is the ElGamal public key of the corresponding voter. This list is delivered to the Voting Authority. In this stage all public system parameters are generated and published, such as P, Q large primes, where $Q|(P-1)$ and $g \in \mathbb{Z}_P^*$ of order Q.

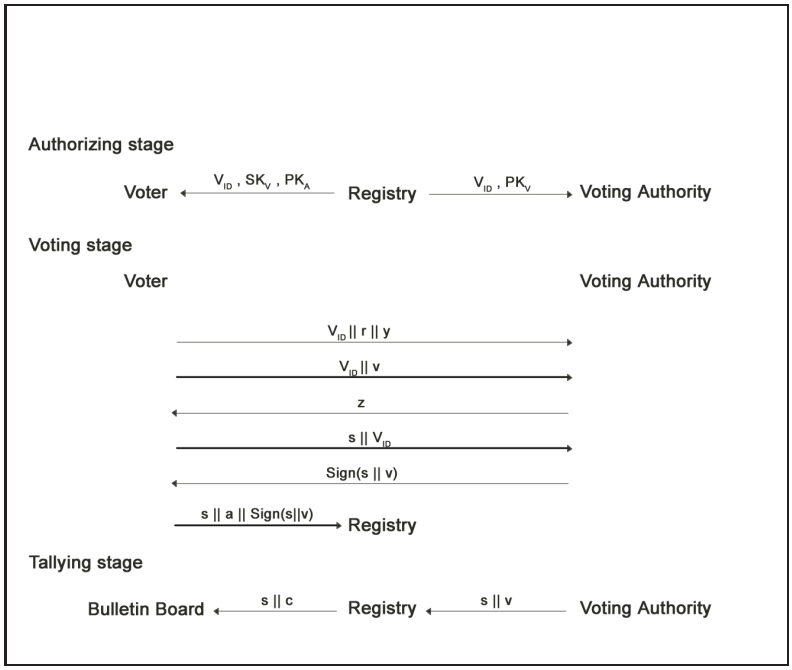

Figure 5.1: The voting scheme

Voting stage

$$\mathcal{V} \longrightarrow \mathcal{A} : E_{PK_A}(V_{ID}||r||y)$$
$$\mathcal{V} \longrightarrow \mathcal{A} : E_{PK_A}(V_{ID}||v)$$
$$\mathcal{A} \longrightarrow \mathcal{V} : E_{PK_V}(z)$$
$$\mathcal{V} \longrightarrow \mathcal{A} : E_{PK_A}(s||V_{ID})$$
$$\mathcal{A} \longrightarrow \mathcal{V} : Sig(s||v)$$
$$\mathcal{V} \longrightarrow \mathcal{R} : E_{PK_R}(s||a||Sig(s||v))$$

The voter chooses random integers a, x and candidate C_i, calculates his ballot with function *vote*, encrypts it with the public key of the Voting Authority and sends it. The voter passes $E_{PK_A}(V_{ID}||v)$ through an anonymous channel. When the Voting Authority receives the message, decrypts it, according to V_{ID} extracts PK_V from the voter roll. Giving PK_V and r to function $ifeligible$ as an input verifies eligibility of the voter. If the voter is eligible for voting it stores all the information, thus the authority can also find out if the voter cast his vote before or not. If the voter is not eligible or has already cast his vote, the Voting Authority bars the voter out of the election.

Voting Authority generates a random integer z, encrypts it with the voter's public key and sends it. After calculating

$$s \equiv x + z \cdot SK_V \pmod{Q}$$

the voter concatenates it with V_{ID} and sends it to the Voting Authority using anonymous channel.

After receiving all the information the Voting Authority looks up PK_V, z, y associated to V_{ID} and runs function $verify$. If it returns 0, then the voter is disclosed from the election otherwise the pair (s, v) is signed and sent back to the voter. After confirming the received signature the voter sends it with the decrypting key a and s to the Registry. If a fraud is detected, then he sends $E_{PK_A}(V_{ID}||r||y)$ through a public channel, $E_{PK_A}(s||v||z||V_{ID})$ through an anonymous channel to the Voting Authority. Voting Authority makes sure of the existence of random parameter z and corresponding values and after applying functions $ifeligible$ and $verify$ sends back $Sig(s||v)$.

Tallying stage

$$\mathcal{A} \longrightarrow \mathcal{R} : E_{PK_R}(s||v)$$
$$\mathcal{R} \longrightarrow \mathcal{BB} : (s||C_i)$$

After the voting phase is finished the Registry receives $(s||v)$ pairs from the Voting Authority, checks validity of the signature received from the voter, computes C_i from v, publishes the pair of (s, C_i) and the relevant voting statistics on $\mathcal{BB}$. In this stage the voter confirms if his vote is correctly listed on $\mathcal{BB}$. If the pair (s, C_i) calculated by the voter is not on $\mathcal{BB}$, then he sends $E_{PK_R}(s||v||a||Sig(s||v))$ through an anonymous channel.

5.3.2 Security analysis

5.3.1 Theorem. *The proposed e-voting scheme is secure, i.e. it satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability and coercion-resistance.*

Proof

Eligibility. During the Authorizing stage a voter is registered only after identifying himself. Only eligible voters receive credential material. Voting Authority ensures eligibility before accepting the ballot by running function $ifeligible$. The Voting Authority cannot impersonate an eligible voter without the official credential issued by the Registry. Therefore, the proposed scheme satisfies eligibility.

Privacy. The vote is encrypted during the process, only in the Tallying stage it is decrypted by the secret key of the Registry. After revealing the votes on $\mathcal{BB}$ and assuming that the Registry and the Voting Authority do not collude, nobody can trace back the identity of the voter.

Unreusability. Each voter possesses different secret key and V_{ID}. If a voter tries to vote with the same credential again the Voting Authority detects it since all the necessary values are stored. Since he cannot generate any other voter's credential, every eligible voter can cast a vote only once.

Fairness. Only in the Tallying stage votes are decrypted, and final results are posted, thus during the voting phase no one has information about any intermediate results.

Robustness. Invalid votes cast by malicious voters are detected in the Tallying stage, after decrypting ballots. These (s, C_i) pairs are marked as invalid by the Registry, or any party can notice them and ask to do it. No coalition of voters can disrupt the election.

Individual verifiability. In the Tallying stage if a voter cannot find the proper (s, C_i) pair on $\mathcal{BB}$ makes a claim. Since the $\mathcal{BB}$ is publicly readable voters can make sure of their own ballots. A voter makes a claim in a way that he shows the signature received and checked in the Voting stage.

Universal verifiability. The final tally and all the votes are listed on $\mathcal{BB}$. Anyone can check the correctness of the results, since $\mathcal{BB}$ is readable by everyone and not erasable or changeable by anyone.

Receipt-freeness, Uncoercibility. The coerced voter V wants to cast vote C, while the adversary $\mathcal{C}$ forces the voter to cast his favorite vote C^*. Voter V calculates the necessary values and functions with value C, follows the steps of the protocol, thus $\mathcal{A}$ accepts v and sends (s, v) to $\mathcal{R}$. At the same time V states to $\mathcal{C}$, he casts vote C^*.

In our scheme

$$View(\mathcal{X} : V) : \{V_{ID}, SK_V, x, a, C^*, z^*, s^*\}.$$

We assume $\mathcal{C}$ generates random integers x, a to V and right after the Authorization stage communicating with V coercer $\mathcal{C}$ is aware of V_{ID}, SK_V. Using anonymous channels $\mathcal{C}$ cannot trace back the message was passed by V to $\mathcal{A}$ or $\mathcal{R}$, in other words even if $\mathcal{C}$ calculates $E_{PK_A}(V_{ID}||v)$, $E_{PK_A}(s||V_{ID})$ and $E_{PK_R}(s||a||Sig(s||v))$ is not able to control if V sent the same messages or not. After the election V chooses an (s', C') pair from $\mathcal{BB}$, where $C' = C^*$ and let $s^* = s'$. It is assumed that the moment when V receives z and calculates s the voter is alone and not being watched, hence V can calculate and state to $\mathcal{C}$ z^*, where

$$s^* \equiv x + z^* \cdot SK_V \pmod{Q}.$$

Since after verifying all the calculations $\mathcal{C}$ accepts $View(\mathcal{X} : V)$, therefore the proposed scheme is receipt-free and uncoercible.

Randomization attack. The randomization attack is prevented, since adversary cannot coerce a voter to cast a different, randomly formed, in-

valid vote. The adversary cannot verify if the coerced voter has cast the prescribed vote or not.

Forced-abstention attack. Even if an adversary can see the voter roll, *i.e.* the list of registered voters, still he is not able to verify if a certain voter has cast a vote or not. Assuming the Voting Authority does not collude with the coercer, the only information he has is on $\mathcal{BB}$. It is not possible to find out the voter from the listed pairs of (s, C_i).

Simulation attack. Even if a voter provides his private keying material (V_{ID}, SK_V) after the Authorizing stage and before the Voting stage, he cannot be coerced by an adversary. An attacker is not able to verify the correctness of the received private keying material.

The proposed scheme satisfies receipt-freeness and protects against randomization, forced-abstention and simulation attack, therefore it is coercion-resistant. □

5.4 A Receipt-Free Homomorphic Election Scheme

5.4.1 The CGS scheme

This scheme was proposed at Eurocrypt'97 by Cramer, Gennaro, and Schoenmarkers [22]. There are s Voting Authorities and m voters participate in the election protocol. The basic CGS protocol offers a choice between two options, that is the following:

- During the set-up procedure all system parameters for robust distributed ElGamal cryptosystem are generated, as described in 4.1.2. Authorities execute key generation protocol due to Pedersen [63], as described in 4.1.2. The result: G_Q subgroup of $\mathbb{Z}_P^*$ with order Q and P, Q, g, h, where $h \equiv g^\alpha \pmod P$.

- A ballot is an ElGamal encryption of the form $(x, y) = (g^k, h^k \cdot G^b)$, where G is a fixed generator of G_Q and $b \in 1, -1$.

- Voter V_i posts ballot (x_i, y_i) to $\mathcal{BB}$ with non-interactive zero knowledge proof of validity.

- After the deadline authorities calculate the product

$$(X, Y) = (\prod_{i=1}^{m} x_i, \prod_{i=1}^{m} y_i).$$

5.4.2 Our scheme

The participants of the protocol are m voters, a Registry $\mathcal{R}$, an authority called Verifier Authority (*VA*) and s Voting Authorities. $\mathcal{R}$ is responsible for managing the Authorizing stage. We do not suppose that $\mathcal{R}$ is honest, $\mathcal{R}$ might collude with adversaries and divulge any information. Verifier Authority (*VA*) manages zero-knowledge proofs of the ballots. *VA* is not expected to be honest, either. After the voting session has completed, s Voting Authorities tally valid votes. Employees of the authorities may also participate as voters. We suppose, all authorities do not collude, there is at least one authority that is honest concerning key generation and message decryption.

Protocol description

The proposed election procedure consists of three distinctive stages: *Authorizing*, *Voting* and *Tallying*.

During the Authorizing stage voters authenticate themselves in person and receive their credentials. All system parameters, sufficient private and public keys are generated. The voter gets his credential in a way that he generates his random reference number, and $\mathcal{R}$ signs it blindly, hence $\mathcal{R}$ cannot connect the credential to the voter. During key-generation $\mathcal{R}$ does not learn anything about private keys either.

During the Voting stage voters create their ballots. Verifier Authority checks eligibility of the voters and if they have already voted before, following it is verified through a non-interactive zero-knowledge proof whether the encrypted ballots sent by the voters are valid or not. This non-interactive zero-knowledge proof is run for a randomized ballot, hence *VA* does not have any information about the form of the encrypted ballot. Voters send their ballots and randomized components authorized by the Verifier Authority to the Bulletin Board. If the ballot appearing on $\mathcal{BB}$ is different or missing, then the voter makes a claim and he can cast his vote again.

During the Tallying stage Voting Authorities calculate the multiplication of valid, encrypted ballots on the bulletin board and divide it with the

product of randomized components. The final results are decrypted and listed.

Building Blocks

The proposed election scheme uses RSA and distributed ElGamal public-key cryptosystems.

At the end of Vote Validation Phase *VA* authorizes the valid ballots using Meta-ElGamal signature scheme [37] with running *SigGenEG* algorithm. V_k verifies with *SigVerEG* whether the received signature is valid.

SigGenEG
Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$. Let denote ESK_{VA} Verifier Authority's ElGamal secure key.
Input: message: $m \in G_Q$
Output: signature: $s_m \in \mathbb{Z}_Q$, $R \in \mathbb{Z}_Q$

1. Verifier Authority chooses random number: $\tilde{k} \in \mathbb{Z}_Q$
2. $R \equiv g^{\tilde{k}} \pmod{P}$
3. $R' \equiv (R \pmod{P}) \pmod{Q}$
4. $m' \equiv (m \pmod{P}) \pmod{Q}$
5. $s_m \equiv ESK_{VA}^{-1}(m' - \tilde{k} \cdot R') \pmod{Q}$

SigVerEG
Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$. Let denote EPK_{VA} Verifier Authority's ElGamal public key.
Input: signature: $s_m \in \mathbb{Z}_Q$, $R \in \mathbb{Z}_Q$, message: m
Output: true, false

1. $R' \equiv (R \pmod{P}) \pmod{Q}$
2. $m' \equiv (m \pmod{P}) \pmod{Q}$

3. Verifies: $EPK_{VA}^{s_m} \cdot R^{R'} \equiv g^{m'} \pmod{P}$

During Vote Validation Phase *VA* authorizes a randomized ballot, this way *VA* cannot connect the ballots being processed during Tallying Stage to ballots that he authorized. Voter V_k generates a proof with *ProofGenEG* for his 'pure' ballots from the randomized ballot signatures sent by *VA*. During Vote Cast Phase V_k sends this proof with his ballots to $\mathcal{BB}$ and anyone is able to verify validity of the ballots with *ProofVerEG* algorithm. *VA* does not learn anything from the values sent to $\mathcal{BB}$: $(\overline{s_m}, \overline{R_m}, \overline{R})$.

ProofGenEG
Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$.
Input: signature: $s_m \in \mathbb{Z}_Q, R \in \mathbb{Z}_Q, \tilde{l} \in \mathbb{Z}_Q$
Output: $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$

1. The voter chooses random number: $\tilde{v} \in \mathbb{Z}_Q$
2. $R' \equiv (R \pmod{P}) \pmod{Q}$
3. $\overline{s_m} \equiv \frac{s_m}{\tilde{l}} \pmod{Q}$
4. $\overline{R} \equiv R^{\frac{\tilde{v}}{\tilde{l}}} \pmod{P}$
5. $\overline{T} \equiv \frac{R'}{\tilde{v}} \pmod{Q}$

ProofVerEG
Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$. Let denote EPK_{VA} Verifier Authority's ElGamal public key.
Input: $m \in \mathbb{Z}_P$, $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$
Output: true, false

1. $m' \equiv (m \pmod{P}) \pmod{Q}$
2. Verifies: $EPK_{VA}^{\overline{s_m}} \cdot \overline{R}^{\overline{T}} \equiv g^{m'} \pmod{P}$

In the following we discuss each step in more details.

Authorizing stage

1. Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$.

2. Voting Authorities generate jointly the public and private keys using distributed ElGamal key generation method (section 4.1.2) in a way, that the private key is not divulged, and the public key is output on $\mathcal{BB}$. Public keys are g and $h \equiv g^K \pmod{P}$, where $K \in \mathbb{Z}_Q$ is the corresponding private key.

3. Registry randomly chooses $v_i \in \mathbb{Z}_Q^*$, $i = 1, \ldots, n$ elements

$$C_i \equiv g^{v_i} \pmod{P}$$

   where C_i represents candidate i from the voter roll and a one-way hash function $M()$ is chosen, v_i, C_i and $M()$ are made public.

4. Let $N_{\mathcal{R}} = P_{\mathcal{R}} \cdot Q_{\mathcal{R}}$, where $P_{\mathcal{R}}$ and $Q_{\mathcal{R}}$ are large primes chosen by the Registry. Let $(RPK_{\mathcal{R}}, N_{\mathcal{R}})$ the RSA public key, such that $(RPK_{\mathcal{R}}, \phi(N_{\mathcal{R}})) = 1$ and $(RSK_{\mathcal{R}}, P_{\mathcal{R}}, Q_{\mathcal{R}})$ the private key, such that $RPK_{\mathcal{R}} \cdot RSK_{\mathcal{R}} \equiv 1 \pmod{\phi}(N_{\mathcal{R}})$. Registry sends its RSA public key to $\mathcal{BB}$.

5. Let $N_{\mathcal{VA}} = P_{\mathcal{VA}} \cdot Q_{\mathcal{VA}}$, where $P_{\mathcal{VA}}$ and $Q_{\mathcal{VA}}$ are large primes chosen by Verifier Authority. Let $(RPK_{\mathcal{VA}}, N_{\mathcal{VA}})$ the RSA public key and $(RSK_{\mathcal{VA}}, P_{\mathcal{VA}}, Q_{\mathcal{VA}})$ the private key, where $(RPK_{\mathcal{VA}}, \phi(N_{\mathcal{VA}})) = 1$, $RPK_{\mathcal{VA}} \cdot RSK_{\mathcal{VA}} \equiv 1 \pmod{\phi}(N_{\mathcal{VA}})$. Verifier Authority generates RSA private and public keys that are being authorized by the Registry, sends the public key to $\mathcal{BB}$.

6. Verifier Authority calculates ElGamal public and private keys, chooses a random $ESK_{VA} \in \mathbb{Z}_Q$ and

$$EPK_{VA} \equiv g^{ESK_{VA}} \pmod{P}.$$

   The private key is ESK_{VA} and the corresponding public key is (EPK_{VA}, P, g).

7. Voters show their identification material to the Registry in person, so the adversary cannot simulate the voter during registration. If a voter has the right to vote, a reference number denoted by $id_k^{\mathcal{R}}$ for the voter V_k is generated by V_k and $\mathcal{R}$ as a join random value, in a way that $\mathcal{R}$ gives only the seed for the pseudorandom number and does not know $id_k^{\mathcal{R}}$. Voter V_k and $\mathcal{R}$ signs blindly in order to authorize V_k's identification number.

By the end of authorizing stage V_k possesses $id_k^{\mathcal{R}}$ and $(M(id_k^{\mathcal{R}}))^{RSK_R} \pmod{N_{\mathcal{R}}}$. All public keys and parameters are on $\mathcal{BB}$:

$$P, Q, g, h, M(), v_i, C_i, RPK_{VA}, N_{VA}, EPK_{VA}, RPK_R, N_{\mathcal{R}}.$$

However the adversary may observe the signing process or collude with $\mathcal{R}$, still cannot learn anything about V_k's reference number or secret key.

Voting stage

The voting stage consists of Vote Validation and Vote Cast phases. Vote Validation phase is a non-interactive zero-knowledge proof based on the idea applied in [22] and [52]. During Vote Validation phase the form of the ballot is proved, i.e. the ballot consists of g^{ϑ} and $h^{\vartheta} \cdot C_i^{(k)}$, where $C_i^{(k)}$ represents candidate i elected by V_k. During the Vote Cast phase the encrypted ballot and the randomized component are sent.

Vote Validation phase

1. The voter V_k first sends

$$id_k^{\mathcal{R}} \pmod{N_{\mathcal{R}}} \,||\, (M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \pmod{N_{\mathcal{R}}}$$

to *VA*. Since during the authorizing stage, due to the randomization, $id_k^{\mathcal{R}}$ and $(M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \pmod{N_{\mathcal{R}}}$ values are not divulged, no one can connect $id_k^{\mathcal{R}}$ to voter V_k. The Verifier Authority checks if the received credential is authorized by the Registry with $\mathcal{R}$'s public key and whether V_k has voted before. If V_k is eligible for voting *VA* and V_k generates a random value, similarly to the Authorizing stage, $id_k^{VA} \pmod{N_{VA}}$ that is an identification value used only in vote validation phase, in order to follow if a voter has already run the zero-knowledge proof. Voter V_k initiates a blind signature algorithm in order to get his identification number authorized and possesses $id_k^{VA} \pmod{N_{VA}}||(M(id_k^{VA}))^{RSK_{VA}} \pmod{N_{VA}}$.

2. V_k sends $id_k^{VA} \pmod{N_{VA}} || (M(id_k^{VA}))^{RSK_{VA}} \pmod{N_{VA}}$ through an anonymous return channel to *VA*. *VA* verifies the signature and if the corresponding voter has not been processed before, sends z_k back through the same channel, where $z_k \in \mathbb{Z}_Q$ random. Since id_k^{VA} signed blindly and anonymous return channel is used, *VA* cannot learn the sender.

3. V_k chooses a candidate i and the corresponding $C_i^{(k)}$ from $\mathcal{BB}$. In order to create his ballot randomly chooses $\alpha_k, \beta_k, \gamma_k \in \mathbb{Z}_Q$ and computes $(G_k, H_k \cdot C_i^{(k)})$ and Y_k where

$$\begin{aligned} G_k &\equiv g^{\alpha_k+\beta_k} \pmod{P} \\ H_k &\equiv h^{\alpha_k+\beta_k} \pmod{P} \\ Y_k &\equiv g^{z_k \cdot \gamma_k} \pmod{P}. \end{aligned}$$

   By randomizing the ballot with β_k, an adversary cannot learn anything from it even if he colludes with *VA*. Y_k plays important role in achieving receipt-freeness.

4. Following V_k runs a non-interactive zero-knowledge proof to prove that he has constructed the ballot correctly, such that he has chosen the value $C_i^{(k)}$ from the voter roll listed on $\mathcal{BB}$. He chooses $r_j, d_j, w_k \in \mathbb{Z}_Q$ random numbers, where $1 \leq j \leq n$ and $j \neq i$, then calculates

$$(\mathsf{A}, \mathsf{B}) = (a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n),$$

   where

$$\begin{aligned} a_i &\equiv g^{w_k} \pmod{P}, \\ b_i &\equiv h^{w_k} \pmod{P}, \end{aligned}$$

   for the elected candidate i and

$$\begin{aligned} a_j &\equiv g^{r_j} \cdot G_k^{d_j} \pmod{P}, \\ b_j &\equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}} \right)^{d_j} \pmod{P} \end{aligned}$$

   for all candidates $j \neq i$.

5. Further, the voter calculates

   $c_k = M(a_1||..||a_n||b_1||..||b_n||G_k||H_k{\cdot}C_i^{(k)}||g||h||id_k^{VA}||(M(id_k^{VA}))^{RSK_{VA}})$

   challenge and

   $$(\mathsf{D},\mathsf{R}) = (d_1, r_1), (d_2, r_2), \ldots, (d_n, r_n)$$

   where for candidate i

   $$\begin{aligned} d_i &\equiv c_k - \sum_{j=1, i\neq j}^{n} d_j \pmod{Q} \\ r_i &\equiv w_k - (\alpha_k + \beta_k) \cdot d_i \pmod{Q}. \end{aligned}$$

6. After calculating all the necessary parameters, V_k chooses a random $\tilde{r} \in \mathbb{Z}_P$ and computes

   $$\tilde{r} \cdot Y_k \pmod{P}.$$

   V_k hides Y_k from *VA* and the adversary.

7. V_k sends the following encrypted randomized ballot and parameters to *VA* through an anonymous return channel:

   $$(\mathsf{A},\mathsf{B})||G_k||H_k \cdot C_i^{(k)}||c_k||(\mathsf{D},\mathsf{R})||id_k^{VA}||(M(id_k^{VA}))^{RSK_{VA}}||\tilde{r} \cdot Y_k.$$

   Since an anonymous return channel is used, *VA* does not know the identity of the sender, *i.e.* *VA* cannot connect the data received through the channel to V_k.

8. After receiving all necessary information *VA* checks whether the voter with id_k^{VA} has already run the zero-knowledge proof, whether id_k^{VA} is signed correctly and calculates the following congruences.

   $$\begin{aligned} c_k &\equiv \sum_{j=1}^{n} d_j \pmod{Q}, \\ a_j &\equiv g^{r_j} \cdot G_k^{d_j} \pmod{P},\ j = 1, \ldots, n \\ b_j &\equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}} \right)^{d_j} \pmod{P},\ j = 1, \ldots, n \end{aligned}$$

If id_k^{VA} is correctly signed and not applied before, then the corresponding voter is eligible for voting and this is his first time to run zero-knowledge proof. If a voter was able to run the zero-knowledge proof several times, then he or she would possess more authorized ballots.

9. If the verification congruences hold, then *VA* signs all the randomized components applying *SigGenEG*. *VA* calculates and sends

$$\begin{aligned} SigGenEG(G_k) &= (s_{m_1}, R_1) \\ SigGenEG(H_k \cdot C_i^{(k)} \cdot Y_k \cdot \tilde{r}) &= (s_{m_2}, R_2) \\ SigGenEG(Y_k \cdot \tilde{r}) &= (s_{m_3}, R_3) \end{aligned}$$

back to the sender through the anonymous return channel.

10. Voter after verifies the three signatures of *VA* with

$$SigVerEG(s_{m_1}, R_1, G_k)$$

$$SigVerEG(s_{m_2}, R_2, H_k \cdot C_i^{(k)} \cdot Y_k \cdot \tilde{r})$$

$$SigVerEG(s_{m_3}, R_3, Y_k \cdot \tilde{r})$$

runs *ProofGenEG* algorithms in order to get authorization of the actual ballots being processed during the Tallying Stage. V_k chooses $\tilde{l_1}, \tilde{l_2}, \tilde{l_3}$ in the following way:

$$\begin{aligned} \tilde{l_1} &\equiv (g^{\beta_k} \pmod P) \pmod Q \\ \tilde{l_2} &\equiv (h^{\beta_k} \cdot \tilde{r} \pmod P) \pmod Q \\ \tilde{l_3} &\equiv (\tilde{r} \pmod P) \pmod Q \end{aligned}$$

and computes

$$ProofGenEG(s_{m_1}, R_1, \tilde{l_1}) = (\overline{s_{m_1}}, \overline{R_1}, \overline{T_1})$$

$$ProofGenEG(s_{m_2}, R_2, \tilde{l_2}) = (\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$$

$$ProofGenEG(s_{m_3}, R_3, \tilde{l_3}) = (\overline{s_{m_3}}, \overline{R_3}, \overline{T_3})$$

Values $(\overline{s_{m_i}}, \overline{R_i}, \overline{T_i})$, where $i = 1, \ldots, 3$ possess a proof of ballot's validity (that is verified by applying *ProofVerEG*), since they are generated from ElGamal signatures (s_{m_i}, R_i) and property of blindness, i.e. these values do not give any information about the ElGamal signature or the ballot itself.

Vote Cast phase

1. Voters send the following information to $\mathcal{BB}$

   $$id_k^{\mathcal{R}} || g^{\alpha_k} || (\overline{s_{m_1}}, \overline{R_1}, \overline{T_1}) || h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k || (\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$$

   through a public channel and

   $$Y_k || (\overline{s_{m_3}}, \overline{R_3}, \overline{T_3})$$

   to *VA* through anonymous channel. The form of the ballot is the ElGamal encryption of $C_i^{(k)} \cdot Y_k \equiv g^{v_i + z_k \cdot \gamma_k} \pmod{P}$, where $z_k \in \mathbb{Z}_Q$ is sent by *VA* through an anonymous channel, hence z_k is not known by the adversary.

2. Voters might check whether their ballots appear on $\mathcal{BB}$. If their ballot is missing or not correct, they can make a claim.

Tallying stage

After the voting stage is over the following computations are made:

1. Verifier Authority runs *ProofVerEG* algorithm for each Y_k and calculates

   $$Y \equiv \prod_{k=1}^{m} Y_k \pmod{P},$$

   where only valid randomized components are considered and sends Y to $\mathcal{BB}$.

2. After verifying validity of encrypted ballots with *ProofVerEG*

   $$\Gamma \equiv \prod_{k=1}^{m} g^{\alpha_k} \pmod{P}$$

   $$\Lambda \equiv \prod_{k=1}^{m} h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k \pmod{P}$$

appear on $\mathcal{BB}$, where only valid ballots are considered.

3. After dividing Λ by Y we get the ElGamal encrypted voting result on $\mathcal{BB}$.

4. Voting Authorities $A_1, A_2, \ldots, A_s$ together calculate the result $C_1^{t_1} \cdot C_2^{t_2} \cdots C_n^{t_n}$ with distributed ElGamal decryption method.

5. Shanks baby step giant step or Pollard rho method might be applied for calculating t_i, $i = 1, \ldots, n$, which gives the election result for candidate i.

Calculation of $t_1, \ldots, t_n$ is considered as a computationally hard problem, it requires $O(m^{(n-1)/2})$ time to get the result.([52]) This scheme can be used for large scale election, if the authorities divide the total value of (Γ, Λ) into parts of reasonable size (e.g. election areas).

5.4.3 Security analysis

5.4.1 Theorem. *The proposed e-voting scheme is secure, i.e. it satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability, receipt-freeness, uncoercibility and protects against randomization and forced-abstention attacks.*

Proof

Eligibility. Verifier Authority checks validity of voters' credentials $id_k^{\mathcal{R}}||(M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}}$ with the corresponding $RPK_{\mathcal{R}}$. If the credential is valid, his $id_k^{\mathcal{R}}$ had been authorized, then the voter's identity material showed in person to Registry was accepted.

Privacy. For encrypting the votes randomized, homomorphic ElGamal public-key cryptosystem is employed, that can be decrypted only, if all authorities collaborate. According to the scheme the voter's vote itself is never decrypted. With the assumption that there is at least one reliable authority, votes remain secret. The vote C_i cannot be derived without knowledge of Y_k. Since during Vote Validation phase all ballots are randomized and cannot be connected to a voter, Verifier Authority does not know how a voter has voted even if *VA* has all information from $\mathcal{BB}$ and zero knowledge proof.

Unreusability. Verifier Authority follows according to the given $id_k^{\mathcal{R}}$ if a voter has casted his valid vote before or not.

Fairness. Determining the tally of the election starts after all the eligible voters have casted their ballots and the votes have been checked if they are valid or not. During the voting stage only the number of eligible voters can be found out.

Robustness. It is detected during the voting phase, if a voter's vote is not valid and only valid votes are considered during the Tallying phase, hence invalid votes cannot distract the elections and it can be also checked if all valid votes are counted. Since all votes are encrypted and they are on $\mathcal{BB}$, authorities or any participant except the voter himself cannot alter votes.

Universal verifiability. After the valid randomized ballots are authorized voters send their encrypted votes on the Bulletin Board. All calculations made on $\mathcal{BB}$, any participant or passive observer can check whether these calculations are correct.

Individual verifiability. The voter himself can check on $\mathcal{BB}$, if his vote has been processed or not. If all public calculations are correct, the result of elections is valid and a voter's vote was made into the final tally as he cast.

Receipt-freeness, Uncoercibility. The proof of receipt-freeness and uncoercibility is based on the fact that there is no enough proof for an adversary how a voter has really voted. An adversary might know a voter's $id_k^{\mathcal{R}}$, $(M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \pmod{N_{\mathcal{R}}}$ and set α_k, γ_k and C_i, v_i, too. During the voting process a voter receives a value z_k and an encrypted ballot

$$Enc_{\alpha_k}(v_i) = (g^{\alpha_k} \pmod P, h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k \pmod P),$$

where $C_i^{(k)} \cdot Y_k = g^{v_i + z_k \cdot \gamma_k}$. Let suppose a coercer has a demand of vote $v_i^* \neq v_i$ and coercer does not know z_k, then the voter is able to cast his vote v_i in a way, that the coercer will accept encrypted ballot on $\mathcal{BB}$. The voter can say the value received form *VA* is

$$z_k^* \equiv \frac{(v_i + z_k \cdot \gamma_k) - v_i^*}{\gamma_k} \pmod Q.$$

Value Y_k never appears on $\mathcal{BB}$ and it is sent during the voting stage through an anonymous channel to *VA* without any identification number or value. *VA* can check its validity, but cannot connect it to a voter. During the Vote Validation phase all data is transported encrypted through an anonymous return channel and no information put on $\mathcal{BB}$.

Randomization attack. If a voter generates randomly formed ballot, it won't be authorized by *VA* during the Vote Validation phase. Only authorized ballots will be considered during the Tallying stage.

Forced-abstention attack. Even Registry does not possess a list of $id_k^{\mathcal{R}}$, since identification numbers are generated by voters and Registry, then they are blindly signed by $\mathcal{R}$, hence an adversary is not able to follow if an eligible voter has voted or not. □

Chapter 6

Appendix

Proof of Lemma 2.4.9.
Let $E = (E_{11} \setminus E_{12}) \cup (E_{21} \setminus E_{22}) \cup E_{33}$ where $t \geq 4$ and

$E_{11} = \{(-a+\varepsilon_1-\varepsilon_2+\varepsilon_3, a-\varepsilon_1+\varepsilon_2, -a+\varepsilon_1, a) \in \mathbb{Z}^4 \mid |a| \leq 4t+10, where$

$(\varepsilon_1, \varepsilon_2, \varepsilon_3) \in \{(-2,2,\varepsilon_3), (2,-2,\varepsilon_3), (-2,1,\varepsilon_3), (2,-1,\varepsilon_3), (-2,0,2), (2,0,-2)\},$
$\varepsilon_3 \in \{-3, ..., 3\}\}$ or
$\varepsilon_1 \in \{-1,0,1\},\ \varepsilon_2 \in \{-2, ..., 2\},\ \varepsilon_3 \in \{-3, ..., 3\}\},$

$E_{12} = \{(-a+\varepsilon_1-\varepsilon_2+\varepsilon_3, a-\varepsilon_1+\varepsilon_2, -a+\varepsilon_1, a) \in \mathbb{Z}^4 \mid$
$\varepsilon_1 = -1, \varepsilon_2 \in \{-2,-1,0\}, \varepsilon_3 \in \{-3, \ldots, 3\}, a = -4t-10,$ or
$\varepsilon_1 = -1, \varepsilon_2 = 1, \varepsilon_3 \in \{-3,-2,-1\}, a = -4t-10,$ or
$\varepsilon_1 = 2, \varepsilon_2 = 0, \varepsilon_3 = -2, a = 4t+10,$ or
$\varepsilon_1 = -2, \varepsilon_2 = 0, \varepsilon_3 = 2, a \leq -4t-9,$ or
$\varepsilon_1 = -2, \varepsilon_2 = 1, \varepsilon_3 \in \{-3, \ldots, 3\}, a \leq -4t-9\},$

$E_{21} = \{(-a+\varepsilon_1-\varepsilon_2+\varepsilon_3, a-\varepsilon_1+\varepsilon_2, -a+\varepsilon_1, a) \in \mathbb{Z}^4 \mid |a| \leq 4t+10,\ where$

$(\varepsilon_1, \varepsilon_2) \in \{(3,-2), (-3,2), (3,-3), (-3,3), (-2,3), (2,-3), (-1,3), (1,-3)\},$
$\varepsilon_3 \in \{-3, ..., 3\}\},$

$E_{22} = \{(-a+\varepsilon_1-\varepsilon_2+\varepsilon_3, a-\varepsilon_1+\varepsilon_2, -a+\varepsilon_1, a) \in \mathbb{Z}^4 \mid$
$\varepsilon_1 = 3, \varepsilon_2 \in \{-2,-3\}, \varepsilon_3 \in \{-3, \ldots, 3\}, a = 4t+10,$ or

$\varepsilon_1 = -3, \varepsilon_2 \in \{2,3\}, \varepsilon_3 \in \{-3,\dots,3\}, a \leq -4t-9$, or
$\varepsilon_1 = -2, \varepsilon_2 = 3, \varepsilon_3 \in \{-3,\dots,3\}, a = -4t-10\}$,

$E_{33} = \{(-4t-6, 4t+8, -4t-10, 4t+11), (4t+6, -4t-8, 4t+10, -4t-11), (-4t-8, 4t+10, -4t-11, 4t+11), (4t+8, -4t-10, 4t+11, -4t-11), (-4t-10, 4t+11, -4t-11, 4t+11), (-4t-11, 4t+11, -4t-11, 4t+11), (4t+11, -4t-11, 4t+11, -4t-11), (4t+10, -4t-11, 4t+11, -4t-11)\}$.

We shall show that the set E satisfies the prerequisites of Theorem 2.1.4 which implies that $P(X) \in \mathcal{C}$.

Let us suppose $a = 0, \varepsilon_1 = \varepsilon_2 = 0, \varepsilon_3 = 1$, then $(1,0,0,0)$ is an element of E. It is clear that $-E \subseteq E$.

Notice that for

$$e = (-a + \varepsilon_1 - \varepsilon_2 + \varepsilon_3, a - \varepsilon_1 + \varepsilon_2, -a + \varepsilon_1, a)$$

we have

$$\tilde{\tau}(e)_4 = -a - \varepsilon_1 - \left\lfloor \frac{s}{p_0} \right\rfloor,$$

where

$$s = t(\varepsilon_2 - \varepsilon_1) + 3\varepsilon_1 + 7\varepsilon_2 + \varepsilon_3 - a.$$

Considering, that $|\varepsilon_2 - \varepsilon_1| \leq 6$ and $|a| \leq 4t + 10$ we can see that applying $\tilde{\tau}$ to $e \in E$ we get the following cases for the fourth component of $\tilde{\tau}(e)$:

a) $-a - \varepsilon_1 - 1$ if $6t + 13 \leq s < 12t + 26$,

b) $-a - \varepsilon_1$ if $0 \leq s < 6t + 13$,

c) $-a - \varepsilon_1 + 1$ if $-6t - 13 \leq s < 0$,

d) $-a - \varepsilon_1 + 2$ if $-12t - 26 \leq s < -6t - 13$.

From here on we prove that $\tilde{\tau}(E) \subseteq E$ by considering several cases.

Case 1: $e \in E_{11} \setminus E_{12}$

We will prove that $\tilde{\tau}(e)_4 \leq 4t + 10$ and $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are from the sets given in E_{11}. Notice that $|\tilde{\tau}(e)_4 + \tilde{\tau}(e)_3| \leq 3$ and $|\tilde{\tau}(e)_3 + \tilde{\tau}(e)_2| \leq 2$.

If $\varepsilon_1 = 1$ and $t > 16$ then $\tilde{\tau}(e) \in E_{11}$, $\forall e \in E_{11}$, because $s = t(\varepsilon_2 - 1) + 3 + 7\varepsilon_2 + \varepsilon_3 - a \leq 6t + 13$, thus only $b)$, $c)$, $d)$ cases should be considered. It is easy to see that $|\tilde{\tau}(e)_4 + \tilde{\tau}(e)_3| \leq 1$. $|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a = 4t + 10$ and $s \geq 0$
2. $a = -4t - 10$ and $s < -6t - 13$

None of the cases occurs since if $a = 4t + 10$ then $s \leq -3t + 7 + \varepsilon_3$, and if $a = -4t - 10$ then $s \geq t - 1 + \varepsilon_3$.

If $\varepsilon_1 = 0$ and $t > 13$ and $\varepsilon_2 > -2$ then $s > -6t - 13$, thus $a)$, $b)$, $c)$ cases are taken into account, $|\tilde{\tau}(e)_4 + \tilde{\tau}(e)_3| \leq 1$, hence $\tilde{\tau}(e) \in E_{11}$. If $\varepsilon_2 = -2$, then $s = -2t - 14 + \varepsilon_3 - a$, and

$$\tilde{\tau}(e) = \begin{cases} (a-2, -a, a, -a) & a \leq -2t - 14 + \varepsilon_3, \\ (a-2, -a, a, -a+1) & -2t - 14 + \varepsilon_3 < a \leq 4t - 1 + \varepsilon_3, \\ (a-2, -a, a, -a+2) & \text{otherwise.} \end{cases}$$

Easy to see that all are elements of E_{11}. For the first two cases: $|\tilde{\tau}(e)_4 + \tilde{\tau}(e)_3| \leq 1$. Considering the last case $\tilde{\tau}(e)_4 + \tilde{\tau}(e)_3 = 2$, $\tilde{\tau}(e)_3 + \tilde{\tau}(e)_2 = 0$ and $\tilde{\tau}(e)_2 + \tilde{\tau}(e)_1 = -2$.

$|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a = 4t + 10$ and $s \geq 6t + 13$
2. $a = -4t - 10$ and $s < 0$
3. $a \leq -4t - 9$ and $s < -6t - 13$

Considering $s = \varepsilon_2 t + 7\varepsilon_2 + \varepsilon_3 - a$ if $a = 4t + 10$ then $s \leq -2t + 4 + \varepsilon_3$, if $a = -4t - 10$ then $s \geq 2t - 4 + \varepsilon_3$ and if $a = -4t - 9$ then $s \geq 2t - 5 + \varepsilon_3$. Hence $|\tilde{\tau}(e)_4| \leq 4t + 10$.

In the following cases we will not detail the value of $\tilde{\tau}(e)_{i+1} + \tilde{\tau}(e)_i$, for $i \in \{1, \ldots, 3\}$. They can be easily calculated like in the cases before.

If $\varepsilon_1 = -1$, then $s > -6t - 13$ for $t > 17$. In case of $a)$, $b)$ $\tilde{\tau}(e) \in E_{11}$. If $-6t - 13 \leq s < 0$, then $\tilde{\tau}(e) = (a + 1 + \varepsilon_2, -a - 1, a, -a + 2) \in E_{11}$.
$|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a = -4t - 10$ and $0 \leq s < 6t + 13$
2. $a \leq -4t - 9$ and $-6t - 13 \leq s < 0$

Considering $s = t(\varepsilon_2 + 1) - 3 + 7\varepsilon_2 + \varepsilon_3 - a$, it is easy to see that the second case cannot occur since if $a \leq -4t - 9$ then $s \geq 3t - 7 + \varepsilon_3$. If $a = -4t - 10$ and $\varepsilon_2 = 2$ or $\varepsilon_2 = 1$ with $\varepsilon_3 \geq -1$ then $s \geq 6t + 13$, otherwise $e \in E_{12}$.

If $\varepsilon_1 = -2$ and $\varepsilon_2 = 2$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = 4t + 8 + \varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a + 4, -a - 2, a, -a + 3) \in E_{21} & a > 4t + 8 + \varepsilon_3, \\ (a + 4, -a - 2, a, -a + 2) \in E_{11} & -2t - 5 + \varepsilon_3 < a \leq 4t + 8 + \varepsilon_3, \\ (a + 4, -a - 2, a, -a + 1) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a = -4t - 10$ and $6t + 13 \leq s$
2. $a \leq -4t - 9$ and $0 \leq s < 6t + 13$
3. $a \leq -4t - 8$ and $-6t - 13 \leq s < 0$

Substituting $a \leq -4t-8$ we get $s \geq 8t+16+\varepsilon_3$, hence $|\tilde{\tau}(e)_4| > 4t+10$ only if the element $e = (4t+6+\varepsilon_3, -4t-6, 4t+8, -4t-10)$. Calculating with the element we get the elements of E_{33}, such that $\tilde{\tau}(e) = (-4t-6, 4t+8, -4t-10, 4t+11), \tilde{\tau}^2(e) = (4t+8, -4t-10, 4t+11, -4t-10) \in E_{11}$. Studying $-e$ we get $\tilde{\tau}(-e) = (4t+6, -4t-8, 4t+10, -4t-10) \in E_{11}$. The negative elements of the path above are also in E_{33}, $\tilde{\tau}(4t + 6, -4t - 8, 4t + 10, -4t - 11) = (-4t - 8, 4t + 10, -4t - 11, 4t + 11)$,
$\tilde{\tau}(-4t-8, 4t+10, -4t-11, 4t+11) = (4t+10, -4t-11, 4t+11, -4t-10) \in E_{11}$,

$\tilde{\tau}(4t+8,-4t-10,4t+11,-4t-11) = (-4t-10,4t+11,-4t-11,4t+11)$,
$\tilde{\tau}(-4t-10,4t+11,-4t-11,4t+11) = (4t+11,-4t-11,4t+11,-4t-10) \in E_{11}$,
$\tilde{\tau}(4t+10,-4t-11,4t+11,-4t-11) = (-4t-11,4t+11,-4t-11,4t+11)$,
$\tilde{\tau}(-4t-11,4t+11,-4t-11,4t+11) = (4t+11,-4t-11,4t+11,-4t-10) \in E_{11}$,
$\tilde{\tau}(4t+11,-4t-11,4t+11,-4t-11) = (-4t-11,4t+11,-4t-11,4t+11)$.

If $\varepsilon_1 = 2$ and $\varepsilon_2 = -2$ and $\varepsilon_3 = \{-3,..,3\}$, then $s = -4t-8+\varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a-4,-a+2,a,-a-2) \in E_{11} & a \leq -4t-8+\varepsilon_3, \\ (a-4,-a+2,a,-a-1) \in E_{11} & -4t-8+\varepsilon_3 < a \leq 2t+5+\varepsilon_3, \\ (a-4,-a+2,a,-a) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t+10$ if

1. $a \geq 4t+9$ and $0 \leq s \leq 6t+13$
2. $a = 4t+10$ and $-6t-13 \leq s \leq 0$

Substituting $a \geq 4t+9$ we get $s \leq -8t-17+\varepsilon_3$, hence none of the cases occurs.

If $\varepsilon_1 = 2$ and $\varepsilon_2 = 0$ and $\varepsilon_3 = -2$, then $s = -2t+4-a$ and

$$\tilde{\tau}(e) = \begin{cases} (a-2,-a+2,a,-a-2) \in E_{11} & a \leq -2t+4, \\ (a-2,-a+2,a,-a-1) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t+10$ if

1. $a \geq 4t+9$ and $0 \leq s$
2. $a = 4t+10$ and $-6t-13 \leq s < 0$

Substituting $a \geq 4t+9$ we get $s \leq -6t-5$, but the element $(-4t-10, 4t+8, -4t-8, 4t+10)$ is in E_{12}.

If $\varepsilon_1 = -2$ and $\varepsilon_2 = 0$ and $\varepsilon_3 = 2$, then $s = 2t - 4 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a+2-a-2, a, -a+2) \in E_{11} & a \leq 2t-4, \\ (a+2-a-2, a, -a+3) \in E_{21} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a \leq -4t - 9$ and $0 \leq s$
2. $a \leq -4t - 8$ and $-6t - 13 \leq s < 0$

Substituting $a \leq -4t - 8$ we get $s \geq 6t + 4$, but the elements $(-a, a + 2, -a - 2, a)$ where $a \leq -4t - 9$ are in E_{12}.

If $\varepsilon_1 = 2$ and $\varepsilon_2 = -1$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = -3t - 1 + \varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a-3, -a+2, a, -a-2) \in E_{11} & a \leq -3t - 1 + \varepsilon_3, \\ (a-3, -a+2, a, -a-1) \in E_{11} & -3t - 1 + \varepsilon_3 < a \leq 3t + 12 + \varepsilon_3, \\ (a-3, -a+2, a, -a) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a \geq 4t + 9$ and $0 \leq s$
2. $a = 4t + 10$ and $-6t - 13 \leq s < 0$

Substituting $a \geq 4t + 9$ we get $s \leq -7t - 10 + \varepsilon_3$.

If $\varepsilon_1 = -2$ and $\varepsilon_2 = 1$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = 3t + 1 + \varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a+3, -a-2, a, -a+1) \in E_{11} & a \leq -3t - 12 + \varepsilon_3, \\ (a+3, -a-2, a, -a+2) \in E_{11} & -3t - 12 + \varepsilon_3 < a \leq 3t + 1 + \varepsilon_3, \\ (a+3, -a-2, a, -a+3) \in E_{21} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a = -4t - 10$ and $6t + 13 \leq s$

2. $a \leq -4t - 9$ and $0 \leq s < 6t + 13$

3. $a \leq -4t - 8$ and $-6t - 13 \leq s < 0$

Substituting $a \leq -4t - 8$ we get $s \geq 7t + 9 + \varepsilon_3$, but the element $(4t + 7 + \varepsilon_3, -4t - 7, 4t + 8, -4t - 10)$ is in E_{22}.

Case 2 $e \in E_{21} \setminus E_{22}$

If $\varepsilon_1 = -1$ and $\varepsilon_2 = 3$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = 4t + 18 + \varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a + 5 - a - 2, a, -a) \in E_{11} & a \leq -2t + 5 + \varepsilon_3, \\ (a + 5 - a - 2, a, -a + 1) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t + 10$ if $a = -4t - 10$ and $s < 6t + 13$, but $s = 8t + 28 + \varepsilon_3$.

If $\varepsilon_1 = 1$ and $\varepsilon_2 = -3$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = -4t - 18 + \varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a - 4, -a + 1, a, -a) \in E_{11} & a \leq 2t - 5 + \varepsilon_3, \\ (a - 4, -a + 1, a, -a + 1) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t + 10$ if $a = -4t - 10$ and $s < -6t - 13$, but $s = -8 + \varepsilon_3$.

If $\varepsilon_1 = 3$ and $\varepsilon_2 = -2$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = -5t - 5 + \varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a - 5, -a + 3, a, -a - 2) \in E_{21} & a \leq t + 8 + \varepsilon_3, \\ (a - 5, -a + 3, a, -a - 1) \in E_{21} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t + 10$ if

1. $a \geq 4t + 9$ and $-6t - 13 \leq s \leq 0$

2. $a = 4t + 10$ and $s < -6t - 13$

Substituting $a \geq 4t+9$ we get $s < -9t-14+\varepsilon_3$, hence we should take only the second case into account, but then the element is $(-4t-5+\varepsilon_3, 4t+5, -4t-7, 4t+10)$ that is in E_{22}.

If $\varepsilon_1 = -3$ and $\varepsilon_2 = 2$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = 5t+5+\varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a+5, -a-3, a, -a+3) \in E_{21} & a > -t-8+\varepsilon_3, \\ (a+5, -a-3, a, -a+2) \in E_{21} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t+10$ if

1. $a \leq -4t-9$ and $6t+13 \leq s$
2. $a \leq -4t-8$ and $0 \leq s < 6t+13$

Substituting $a \leq -4t-8$ we get $s \geq 9t+13+\varepsilon_3$, but the elements $(-a-5+\varepsilon_3, a+5, -a-3, a)$ where $a \leq -4t-9$ are in E_{22}.

If $\varepsilon_1 = -3$ and $\varepsilon_2 = 3$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = 6t+12+\varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a+6, -a-3, a, -a+2,) \in E_{21} & a \leq -1+\varepsilon_3, \\ (a+6, -a-3, a, -a+3) \in E_{21} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t+10$ if

1. $a \leq -4t-9$ and $6t+13 \leq s$
2. $a \leq -4t-8$ and $0 \leq s < 6t+13$

Substituting $a \leq -4t-8$ we get $s \geq 10t+20+\varepsilon_3$, but the elements $(-a-6+\varepsilon_3, a+6, -a-3, a)$ where $a \leq -4t-9$ are in E_{22}.

If $\varepsilon_1 = 3$ and $\varepsilon_2 = -3$ and $\varepsilon_3 = \{-3, .., 3\}$, then $s = -6t-12+\varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a-6, -a+3, a, -a-2) \in E_{21} & a \leq 1+\varepsilon_3, \\ (a-6, -a+3, a, -a-1) \in E_{21} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t+10$ if

1. $a \geq 4t+9$ and $-6t-13 \leq s \leq 0$
2. $a = 4t+10$ and $s < -6t-13$

Substituting $a \geq 4t+9$ we get $s \leq -10t-21+\varepsilon_3$, but the element $(-4t-4+\varepsilon_3, 4t+4, -4t-7, 4t+10,)$ is in E_{22}.

If $\varepsilon_1 = -2$ and $\varepsilon_2 = 3$ and $\varepsilon_3 = \{-3,..,3\}$, then $s = 5t+15+\varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a+5, -a-2, a, -a+1) \in E_{11} & a \leq -t+2+\varepsilon_3, \\ (a+5, -a-2, a, -a+2) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t+10$ if

1. $a = -4t-10$ and $6t+13 \leq s$
2. $a \leq -4t-9$ and $0 \leq s < 6t+13$

Substituting $a \leq -4t-9$ we get $s \geq 9t+24+\varepsilon_3$, but the $(4t+5+\varepsilon_3, -4t-5, 4t+8, -4t-10)$ element is in E_{22}.

If $\varepsilon_1 = 2$ and $\varepsilon_2 = -3$ and $\varepsilon_3 = \{-3,..,3\}$, then $s = -5t-15+\varepsilon_3 - a$ and

$$\tilde{\tau}(e) = \begin{cases} (a-5, -a+2, a, -a-1) \in E_{11} & a \leq t-2+\varepsilon_3, \\ (a-5, -a+2, a, -a) \in E_{11} & \text{otherwise.} \end{cases}$$

$|\tilde{\tau}(e)_4| > 4t+10$ only if $a = 4t+10$ and $s \geq -6t-13$. Substituting $a = 4t+10$ we get $s \leq -9t-25+\varepsilon_3$.

Case 3 There are no images of other elements of E in $E_{12} \bigcup E_{22}$

$(-a-\varepsilon_2+\varepsilon_3, a+\varepsilon_2, -a-1, a)$ where $\varepsilon_2 \in \{-1,0,1\}$ and $\varepsilon_3 \in \{-3,...,3\}$ and $a = -4t-10$.

$$(-a-\varepsilon_2+\varepsilon_3, a+\varepsilon_2, -a-1, a) = \begin{cases} \tilde{\tau}(a+\varepsilon_2-\varepsilon_3+\varepsilon_4, -a-\varepsilon_2+\varepsilon_3, a+\varepsilon_2, -a-1) \\ -(a+\varepsilon_2-\varepsilon_3, -a-\varepsilon_2, a+1, -a). \end{cases}$$

In the first case $s = t(\varepsilon_3 - \varepsilon_2) + 3\varepsilon_2 + 7\varepsilon_3 + \varepsilon_4 - 3t - 12$

$$\tilde{\tau}(a+\varepsilon_2-\varepsilon_3+\varepsilon_4, -a-\varepsilon_2+\varepsilon_3, a+\varepsilon_2, -a-1)_4 = \begin{cases} a+2-\varepsilon_2 & 0 \le s, \\ a+3-\varepsilon_2 & -6t-13 \le s < 0, \\ a+4-\varepsilon_2 & s < -6t-13. \end{cases}$$

Since $\varepsilon_2 < 3$ the only case we can get a as a first coordinate, if $\varepsilon_2 = 2$, but then $s \le 0$.

In the second case $s = t(\varepsilon_2 - \varepsilon_3) - 3\varepsilon_2 - 7\varepsilon_3 - \varepsilon_4 + 3t + 12$

$$\tilde{\tau}(-a-\varepsilon_2+\varepsilon_3-\varepsilon_4, a+\varepsilon_2-\varepsilon_3, -a-\varepsilon_2, a+1)_4 = \begin{cases} -a-3+\varepsilon_2 & 6t+13 \le s, \\ -a-2+\varepsilon_2 & 0 \le s < 6t+13, \\ -a-1+\varepsilon_2 & s < 0. \end{cases}$$

Since $\varepsilon_2 < 3$ we can get $-a$ as a fourth coordinate, if $\varepsilon_2 = 1$, but then $s > 0$ or $\varepsilon_2 = 2$ if $\varepsilon_3 \ge 0$.

$(-a+3-\varepsilon_2+\varepsilon_3, a-3+\varepsilon_2, -a+3, a)$ where $\varepsilon_2 \in \{-2,-3\}$ and $\varepsilon_3 \in \{-3,...,3\}$ and $a = 4t+10$.
$(-a+3-\varepsilon_2+\varepsilon_3, a-3+\varepsilon_2, -a+3, a) =$

$$\begin{cases} \tilde{\tau}(a-3+\varepsilon_2-\varepsilon_3+\varepsilon_4, -a+3-\varepsilon_2+\varepsilon_3, a-3+\varepsilon_2, -a+3) \\ -(a-3+\varepsilon_2-\varepsilon_3, -a+3-\varepsilon_2, a-3, -a). \end{cases}$$

Hence $s = t(\varepsilon_3-\varepsilon_2)+3\varepsilon_2+7\varepsilon_3+\varepsilon_4+4t+7$ and $3t-20+\varepsilon_4 \le s \le 10t+19+\varepsilon_4$
$\tilde{\tau}(a-3+\varepsilon_2-\varepsilon_3+\varepsilon_4, -a+3-\varepsilon_2+\varepsilon_3, a-3+\varepsilon_2, -a+3)_4 =$

$$\begin{cases} a-3-\varepsilon_2-1 & 6t+13 \le s \le 12t+26, \\ a-3-\varepsilon_2 & 0 \le s < 6t+13. \end{cases}$$

Since $\varepsilon_2 \ne 4$, and even if $\varepsilon_2 = -3$, then $\varepsilon_3 \in \{-1,-2,-3\}$ should happen, but there are no elements with these properties in E.

In the second case $s = t(-\varepsilon_3+\varepsilon_2) - 3\varepsilon_2 - 7\varepsilon_3 - \varepsilon_4 - 4t - 7$, $\tilde{\tau}(-a+3-\varepsilon_2+\varepsilon_3-\varepsilon_4, a-3+\varepsilon_2-\varepsilon_3, -a+3-\varepsilon_2, a-3)_4 =$

$$\begin{cases} -a+3+\varepsilon_2+1 & -6t-13 \le s \le 0, \\ -a+3+\varepsilon_2+2 & -12t-26 \le s < -6t-13. \end{cases}$$

Since $\varepsilon_2 > -4$, none of the cases can occur.

$(a, -a-3, a+3+\varepsilon_2, -a+3-\varepsilon_2+\varepsilon_3)$ where $\varepsilon_2 \in \{2,3\}$ and $\varepsilon_3 \in \{-3,...,3\}$ and $a \leq -4t-9$.
$(-a-3-\varepsilon_2+\varepsilon_3, a+3+\varepsilon_2, -a-3, a) =$

$$\begin{cases} \tilde{\tau}(a+3+\varepsilon_2-\varepsilon_3+\varepsilon_4, -a-3-\varepsilon_2+\varepsilon_3, a+3+\varepsilon_2, -a-3) \\ -(a+3+\varepsilon_2-\varepsilon_3, -a-3-\varepsilon_2, a+3, -a). \end{cases}$$

Since $s = t(\varepsilon_3-\varepsilon_2)+3\varepsilon_2+7\varepsilon_3+\varepsilon_4+a+3$ and $-10t-19+\varepsilon_4 \leq s \leq -3t+21+\varepsilon_4$ $\tilde{\tau}(a+3+\varepsilon_2-\varepsilon_3+\varepsilon_4, -a-3-\varepsilon_2+\varepsilon_3, a+3+\varepsilon_2, -a-3)_4 =$

$$\begin{cases} a+3-\varepsilon_2+1 & -6t-13 \leq s < 0, \\ a+3-\varepsilon_2+2 & -12t-26 \leq s < 6t+13. \end{cases}$$

Since $\varepsilon_2 < 4$, none of the cases can occur.

In the second case $s = t(-\varepsilon_3+\varepsilon_2)-3\varepsilon_2-7\varepsilon_3+\varepsilon_4-a-3$, $\tilde{\tau}(-a-3-\varepsilon_2+\varepsilon_3-\varepsilon_4, a+3+\varepsilon_2-\varepsilon_3, -a-3-\varepsilon_2, a+3)_4 =$

$$\begin{cases} -a-3+\varepsilon_2-1 & 6t+13 \leq s \leq 12t+26, \\ -a-3+\varepsilon_2 & 0 \leq s < 6t+13. \end{cases}$$

Since $\varepsilon_2 \neq 4$, and even if $\varepsilon_2 = 3$, then $\varepsilon_3 \in \{1,2,3\}$ should happen, but there are no elements with these properties in E.

$(a, -a-2, a+2+\varepsilon_2, -a-2-\varepsilon_2+\varepsilon_3)$ where $\varepsilon_2 \in \{0,1,3\}$ and $\varepsilon_3 \in \{-3,...,3\}$ and $a \leq -4t-9$.
$(-a-2-\varepsilon_2+\varepsilon_3, a+2+\varepsilon_2, -a-2, a) =$

$$\begin{cases} \tilde{\tau}(a+2+\varepsilon_2-\varepsilon_3+\varepsilon_4, -a-2-\varepsilon_2+\varepsilon_3, a+2+\varepsilon_2, -a-2) \\ -(a+2+\varepsilon_2-\varepsilon_3, -a-2-\varepsilon_2, a+2, -a). \end{cases}$$

Since $s = t(\varepsilon_3-\varepsilon_2)+3\varepsilon_2+7\varepsilon_3+\varepsilon_4+a+2$ and $-10t-20+\varepsilon_4 \leq s \leq -t+14+\varepsilon_4$ $\tilde{\tau}(a+2+\varepsilon_2-\varepsilon_3+\varepsilon, -a-2-\varepsilon_2+\varepsilon_3, a+2+\varepsilon_2, -a-2)_4 =$

$$\begin{cases} a+2-\varepsilon_2, t=17 & 0 \leq s < 6t+13, \\ a+2-\varepsilon_2+1 & -6t-13 \leq s < 0, \\ a+2-\varepsilon_2+2 & -12t-26 \leq s < 6t+13. \end{cases}$$

Since $\varepsilon_2 \neq 4$ and $\varepsilon_2 \neq 2$, and if $\varepsilon_2 = 3$, then $\varepsilon_3 \in \{1, 2, 3\}$ should be true, but there are no elements with these properties in E.

In the second case $s = t(-\varepsilon_3 + \varepsilon_2) - 3\varepsilon_2 - 7\varepsilon_3 + \varepsilon_4 - a - 2$, $\tilde{\tau}(-a - 2 - \varepsilon_2 + \varepsilon_3 - \varepsilon_4, a + 2 + \varepsilon_2 - \varepsilon_3, -a - 2 - \varepsilon_2, a + 2)_4 =$

$$\begin{cases} -a - 2 + \varepsilon_2 - 1 & 6t + 13 \leq s \leq 12t + 26, \\ -a - 2 + \varepsilon_2 & 0 \leq s < 6t + 13. \end{cases}$$

Since $\varepsilon_2 \neq 2$, and even if $\varepsilon_2 = 3$, then $\varepsilon_3 \in \{0, -1, -2, -3\}$ should happen, but there are no elements with these properties in E.

$(-a, a - 2, -a + 2, a)$ where $a = 4t + 10$.

$$(-a, a - 2, -a + 2, a) = \begin{cases} \tilde{\tau}(a + \varepsilon_3, -a, a - 2, -a + 2) \\ -(a, -a + 2, a - 2, -a). \end{cases}$$

In the first case $s = 2t - 6 + \varepsilon_3$ and $\tilde{\tau}(a + \varepsilon_3, -a, a - 2, -a + 2)_4 = a - 2$, in the second case $s = -2t + 6 + \varepsilon_3$ and $\tilde{\tau}(-a + \varepsilon_3, a, -a + 2, a - 2)_4 = -a + 3$.

To prove that for every $e \in E$ there exists some $l > 0$ with $\tilde{\tau}^l(e) = 0$ we show that applying the mapping $\tilde{\tau}$ to any element $e = (-a + \varepsilon_1 - \varepsilon_2 + \varepsilon_3, a - \varepsilon_1 + \varepsilon_2, -a + \varepsilon_1, a)$, where $|a| \leq 4t + 10$ we get a *spiral*. A spiral is a $\tilde{\tau}$ sequence of elements where the ε_i, $i = 1, 2, 3$ of the first and the last elements are the same and the first coordinate of the last element is smaller in absolute value then the one of first one. We will denote a spiral by $(a_1, a_2, a_3, a_4); a_5, \ldots, a_n$. There is a spiral for any $a \in [-4t - 10, 4t + 10]$ and ε_i. It means that once a sequence arrives at a spiral then it will follow it and it will be decreasing in the first coordinate in absolute value until it arrives to zero or it turns into another spiral.

There are 14 spirals:

- $(-a - 5, a + 4, -a - 2, a); -a + 3, a - 5, -a + 7, a - 8, -a + 10, a - 12$
- $(-a + 6, a - 4, -a + 2, a); -a - 1, a + 3, -a - 5, a + 7$
- $(-a + 6, a - 4, -a + 2, a); -a - 1, a + 2, -a - 3, a + 5, -a - 7, a + 9$
- $(-a + 4, a - 2, -a + 1, a); -a - 1, a + 3, -a - 4, a + 5$

- $(-a+3, a-2, -a+1, a); -a, a+1, -a-2, a+3$
- $(-a, a, -a, a); -a+1, a-2, -a+2, a-2, -a+2$
- $(-a, a, -a, a); -a+1, a-1, -a+1, a-1$
- $(-a+3, a-2, -a+1, a); -a-1, a+2$, where $a < 0$
- $(-a-2, a+1, -a, a); -a, a+1, -a-2, a+3, -a-3$
- $(-a+1, a, -a, a); -a+1, a-1, -a+1$
- $(-a-2, a+2, -a-1, a); -a+2, a-3, -a+3, a-2, -a+1$
- $(-a-2, a+1, -a, a); -a, a, -a+1, a-2, -a+3, a-3$
- $(-a+3, a-2, -a+1, a); -a-1, a+3, -a-4, a+5, -a-5, a+6, -a-7, a+8$
- $(-a+4, a-2, -a+1, a); -a, a, -a+1, a-2, -a+4, a-5, -a+6$

If $\varepsilon_1 = 1$, then $e = (-a+1-\varepsilon_2+\varepsilon_3, a-1+\varepsilon_2, -a+1, a)$.

$$\tilde{\tau}(e) = \begin{cases} (a-1+\varepsilon_2, -a+1, a, -a-1), & 0 \le s < 6t+13, \\ (a-1+\varepsilon_2, -a+1, a, -a), & -6t-13 \le s < 0 \\ (a-1+\varepsilon_2, -a+1, a, -a+1), & -12t-26 \le s < -6t-13 \end{cases}$$

In the first case $a \le t(\varepsilon_2 - 1) + 3 + 7\varepsilon_2 + \varepsilon_3$, so $a \le t + 17 + \varepsilon_3$.

$$\tilde{\tau}^2(e) = \begin{cases} (-a+1, a, -a-1, a+2), & 0 \le 2t+5+\varepsilon_2+a < 6t+13, \\ (-a+1, a, -a-1, a+3), & -6t-13 \le 2t+5+\varepsilon_2+a < 0 \end{cases}$$

Considering the first case again, since $-2t-5-\varepsilon_2 \le a \le t+17+\varepsilon_3$

$$\tilde{\tau}^3(e) = \begin{cases} (a, -a-1, a+2, -a-3), & 0 \le -2t-5-a < 6t+13, \\ (a, -a-1, a+2, -a-2), & -6t-13 \le -2t-5-a < 0 \end{cases}$$

In the first case $s = 2t+6+a$, so $-1 \leq s \leq 3t+23+\varepsilon_3$. If $a = -2t-7$, then $\tilde{\tau}^4(e) = (-a-1, a+2, -a-3, a+5)$, otherwise $(-a-1, a+2, -a-3, a+4)$, $\varepsilon_i = (1, -1, 1)$. There is a spiral starting with $(-a+1, a, -a-1, a+2)$, where $\varepsilon_i = (1, -1, 1)$.

In a similar way we can find spirals for all other cases.□

Summary

The present dissertation consists of two more or less independent topics. First part of this work deals with generalized number systems, especially Canonical Number Systems and Symmetric Shift Radix Systems. The second part belongs to cryptographically secure electronic elections. Two new voting schemes are presented: a coercion-resistant voting scheme based on blind signatures and a receipt-free homomorphic election scheme.

Generalized Number Systems

First chapter contains the historical background, the presentation overview and our main results. In the second chapter we deal with **Canonical Number Systems**. CNS bases are explicitly known for some quadratic, cubic and quartic fields ([43],[44],[30],[33],[86],[8],[49],[5],[67]). Our main result is the characterization of CNS bases in algebraic number fields including quartic cyclotomic fields, simplest quartic fields and two families of orders in quartic number fields. The results of this chapter are contained in our paper [17]. This paper is a joint work with Horst Brunotte and Attila Pethő.
In the sequel we denote by $\mathbb{Q}$ the field of rational numbers, by $\mathbb{Z}$ the set of integers and by $\mathbb{N}$ the set of nonnegative integers. For an algebraic integer γ we let $\mu_\gamma \in \mathbb{Z}[X]$ be its minimal polynomial and $\mathcal{C}_\gamma$ the set of all CNS bases for $\mathbb{Z}[\gamma]$.

6.0.2 Definition. Let $P(X) = X^d + p_{d-1}X^{d-1} + \cdots + p_1X + p_0 \in \mathbb{Z}[X]$, $N = \{0, 1, \ldots, |p_0| - 1\}$ and $\mathcal{R} := \mathbb{Z}[X]/P(X)\mathbb{Z}[X]$ and denote the image of X under the canonical epimorphism from $\mathbb{Z}[X]$ to $\mathcal{R}$ by x. If every non-zero element $A(x) \in \mathcal{R}$ can be written uniquely in the form $A(x) = a_0 + a_1x + \cdots + a_lx^l$ with $a_0, \ldots, a_l \in N, a_l \neq 0$, we call (P, N) a

canonical number system (CNS for short). $P(X)$ is called CNS polynomial, to N we refer as the set of digits.

We denote by $\mathcal{C}$ the set of CNS polynomials, and α is a CNS basis for $\mathbb{Z}[\alpha]$ if and only if μ_α is a CNS polynomial.

6.0.3 Theorem. *Let γ be an algebraic integer. Then there exist finite effectively computable disjoint subsets $\mathcal{F}_0(\gamma), \mathcal{F}_1(\gamma) \subset \mathcal{C}_\gamma$ with the properties:*

(i) For every $\alpha \in \mathcal{C}_\gamma$ there exists some $n \in \mathbb{N}$ with $\alpha + n \in \mathcal{F}_0(\gamma) \cup \mathcal{F}_1(\gamma)$.

(ii) $\mathcal{F}_1(\gamma)$ consists of fundamental CNS bases for $\mathbb{Z}[\gamma]$.

For finding CNS bases a modified version of the algorithm given by B. Kovács and A. Pethő [49] is applied. This algorithm assumes existence of a set, that contains representatives of the equivalence classes of generators of power integral bases of the given order and finds sets $\mathcal{F}_0(\gamma)$ and $\mathcal{F}_1(\gamma)$.

Now we will treat the cyclotomic fields of degree 4.

6.0.4 Theorem. *Let $\zeta_5, \zeta_8, \zeta_{12}$ be a primitive fifth, eighth and twelfth root of unity respectively. Then we have $\mathcal{F}_0(\mathbb{Q}(\zeta_i)) = \emptyset$ for $i \in \{5, 8, 12\}$ and*
$\mathcal{F}_1(\mathbb{Q}(\zeta_5)) = \{-2 + \zeta_5, -3 - \zeta_5, -2 + \zeta_5 + \zeta_5^3, -3 - \zeta_5 - \zeta_5^3\}$.
$\mathcal{F}_1(\mathbb{Q}(\zeta_8)) = \{-3 \pm \zeta_8^k \mid k = 1, 3, 5, 7\}$.
$\mathcal{F}_1(\mathbb{Q}(\zeta_{12})) = \{-3 + \zeta_{12}, -3 - \zeta_{12}, -3 + \zeta_{12}^{-1}, -3 - \zeta_{12}^{-1}, -1 - \zeta_{12}^2 + \zeta_{12}^{-1}, -2 + \zeta_{12}^2 - \zeta_{12}^{-1}\}$.

For $t \in \mathbb{Z} \setminus \{0, \pm 3\}$ let $P_t(X) = X^4 - tX^3 - 6X^2 + tX + 1$. Let $\vartheta = \vartheta_t$ be a root of $P_t(X)$, then the infinite parametric family of number fields $K_t = K = \mathbb{Q}(\vartheta_t)$ is called *simplest quartic fields*. P. Olajos [57] proved that K_t admits a power integral bases if and only if $t = 2$ and $t = 4$, moreover he found all generators of power integral bases in these fields. Using his result we are able to compute all CNS bases in such fields.

6.0.5 Theorem. *We have $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ and $\mathcal{F}_1(\mathbb{Q}(\vartheta_2)) = \mathcal{G}_2$ and $\mathcal{F}_1(\mathbb{Q}(\vartheta_4)) = \mathcal{G}_4$ where $\mathcal{G}_2$ and $\mathcal{G}_4$ are explicitly given.*

6.0.6 Remark. For detailed description of sets $\mathcal{G}_2$ and $\mathcal{G}_4$ we refer to section 2.4.

Power integral bases in the polynomial order $\mathbb{Z}[\alpha]$ of K_t were described by G. Lettl and A. Pethő [53].

6.0.7 Theorem. *Let $t \in \mathbb{N} \setminus \{0,3\}$ and ϑ denote a root of the polynomial $X^4 - tX^3 - 6X^2 + tX + 1$. Then we have $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ and $\mathcal{F}_1(\mathbb{Q}(\vartheta)) = \mathcal{G} \cup \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_4$ where $\mathcal{G}$, $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_4$ are explicitly given.*

6.0.8 Remark. For detailed description of sets $\mathcal{G}$, $\mathcal{G}_1$, $\mathcal{G}_2$ and $\mathcal{G}_4$ we refer to section 2.4.

Finally we consider a family of orders in a parameterized family of quartic number fields, where all power integral bases are known. Let $t \in \mathbb{Z}$, $t \geq 0$, and $P(X) = X^4 - tX^3 - X^2 + tX + 1$. Denote by α one of the zeros of $P(X)$. In the following we deal with the order $\mathcal{O} = Z[\alpha]$ of $\mathbb{Q}(\alpha)$. Based on paper of M. Mignotte, A. Pethő and R. Roth [55] we give the following result.

6.0.9 Theorem. *Let $t \geq 4$. We have $\mathcal{F}_0(\mathbb{Q}(\alpha)) = \emptyset$ and $\mathcal{F}_1(\mathbb{Q}(\alpha)) = \mathcal{G}_4 \cup \mathcal{G}_t$ where*

$$\begin{aligned}\mathcal{G}_4 &= \left\{209\alpha + 140\alpha^2 - 49\alpha^3 + 350, 209\alpha - 312\alpha^2 + 64\alpha^3 - 71\right\}\\ \mathcal{G}_t &= \left\{\alpha + t + 1, \alpha + t\alpha^2 - \alpha^3 + t + 2, t\alpha + (t-1)\alpha^2 - \alpha^3 + 8,\right.\\ &\quad \left. t\alpha - (t+1)\alpha^2 + \alpha^3 + 2, \alpha - \alpha^3 + 2, \alpha - t(t^2+1)\alpha^2 + t^2\alpha^3 - t + 1\right\}.\end{aligned}$$

Chapter three is devoted to **Symmetric Shift Radix Systems**. Two dimensional SSRS is treated in [9] by Akiyama and Scheicher, we will deal with three-dimensional SSRS.

The results of this chapter are based on [40], that is a joint work with Klaus Scheicher, Paul Surer and Jörg M. Thuswaldner.

6.0.10 Definition. (***cf.*** **[9]**) Let $d \geq 1$ be an integer, $\mathbf{r} \in \mathbb{R}^d$, and let

$$\tau_{\mathbf{r}} : \mathbb{Z}^d \to \mathbb{Z}^d, \quad \mathbf{a} = (a_1, \ldots, a_d) \mapsto \left(a_2, \ldots, a_d, -\left\lfloor \mathbf{ra} + \frac{1}{2} \right\rfloor\right). \tag{6.0.1}$$

Then $\tau_{\mathbf{r}}$ is called a *symmetric shift radix system* (SSRS for short), if $\forall \mathbf{a} \in \mathbb{Z}^d \quad \exists n \in \mathbb{N} : \tau_{\mathbf{r}}^n(\mathbf{a}) = \mathbf{0}$.

Let

$$\mathcal{D}_d := \left\{ \mathbf{r} \in \mathbb{R}^d \,\middle|\, \forall \mathbf{a} \in \mathbb{Z}^d \, \exists n, l \in \mathbb{N} : \tau_{\mathbf{r}}^k(\mathbf{a}) = \tau_{\mathbf{r}}^{k+l}(\mathbf{a}) \; \forall k \geq n \right\} \text{ and}$$
$$\mathcal{D}_d^0 := \left\{ \mathbf{r} \in \mathbb{R}^d \,|\tau_{\mathbf{r}} \text{ is an SSRS} \right\}.$$

As a new result we prove that $\mathcal{D}_3^0$ is an union of four polyhedra and a polygon, by employing the algorithm that is established for SSRS in [9]. In [9] it has been shown that $\mathcal{E}_d(1) \subset \mathcal{D}_d \subset \overline{\mathcal{E}_d(1)}$. Let a non-zero period $\pi := (a_1, \ldots, a_d); a_{d+1}, \ldots, a_L$ be given. We may ask for the set $P(\pi)$ of all $\mathbf{r} \in \mathcal{D}_d$ for that π occurs as a period of $\tau_{\mathbf{r}}$. By the definition of $\tau_{\mathbf{r}}$, an element $\mathbf{r} \in P(\pi)$ has to satisfy the system of L double inequalities

$$-\frac{1}{2} \leq r_1 a_{1+i} + r_2 a_{2+i} + \cdots + r_d a_{d+i} + a_{d+1+i} < \frac{1}{2}. \tag{6.0.2}$$

Here i runs from 0 to $L-1$ and $a_{L+1} = a_1, \ldots, a_{L+d} = a_d$. Such a system characterizes a convex polyhedron, which is possibly degenerated or equal to the empty set. Therefore we will call $P(\pi)$ a *cutout polyhedron*. Since each point $\mathbf{r} \in P(\pi)$ has π as a period of the associated mapping $\tau_{\mathbf{r}}$ the set $P(\pi)$ has empty intersection with $\mathcal{D}_d^0$. Thus we get the representation

$$\mathcal{D}_d^0 = \mathcal{D}_d \setminus \bigcup_{\pi \neq \mathbf{0}} P(\pi),$$

where the union is extended over all non-zero periods π. Since the set of periods is infinite, this expression is not suitable for calculations. The following theorem shows how to reduce the set of possible periods to a finite set and gives an efficient algorithm for a closed subset H of $\operatorname{int} \mathcal{D}_d = \mathcal{E}_d(1)$ to determine $H \cap \mathcal{D}_d^0$.

6.0.11 Theorem. (*cf.* [9]) *Let $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k \in \mathcal{D}_d$ and let $H := \square(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k)$ be the convex hull of $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_k$. Assume that $H \subset \operatorname{int} \mathcal{D}_d$ and sufficiently small in diameter. Then there exists an algorithm to construct a finite directed graph $G(H) = V \times E$ with vertices $V \subset \mathbb{Z}^d$ and edges $E \subset V \times V$ which satisfies*

1. *$\pm \boldsymbol{e}_i \in V$ for all $i = 1, \ldots, d$,*

2. *$\mathcal{G}(\mathcal{V}(\boldsymbol{x}))$ is a subgraph of $G(H)$ for all $\boldsymbol{x} \in H$,*

3. *$H \cap \mathcal{D}_d^0 = H \setminus \bigcup_\pi P(\pi)$, where π runs through all periods induced by the nonzero primitive cycles of G.*

Our aim is to characterize $\mathcal{D}_3^0$. From [78, 84] we calculate $\mathcal{E}_3(1) = \left\{(x,y,z) \in \mathbb{R}^3 \,\middle|\, |x|<1, |y-xz|<1-x^2, |x+z|<|y+1|\right\}$. Let $\mathcal{E}_3' := \left\{(x,y,z) \in \mathbb{R}^3 \middle|\, |x| \le 1 \wedge |y-xz| \le 1-x^2 \wedge |x+z| \le |y+1| \wedge |y-1| \le 2 \wedge |z| \le 3\right\}$ and consider the intersection of $\mathcal{E}_3'$ with the hyperplane $A_c := \left\{(x,y,z) \in \mathbb{R}^3 \mid x-c=0\right\}$ for constant c.

6.0.12 Lemma. *For any $|c|<1$ the intersection of $\mathcal{E}_3'$ with the plane A_c yields the closed triangle $\triangle(A_c^{(1)}, A_c^{(2)}, A_c^{(3)})$ with $A_c^{(1)} = (c,-1,-c), A_c^{(2)} = (c, 1-2c, c-2), A_c^{(3)} = (c, 2c+1, c+2)$.*

6.0.13 Theorem. $\overline{\mathcal{E}_3(1)} = \mathcal{E}_3'$.

The number of inequalities can be reduced, we gain

$$\overline{\mathcal{E}_3(1)} = \left\{(x,y,z) \,\middle|\, |x+z| \le 1+y \wedge y-xz \le 1-x^2 \wedge |z| \le 3\right\}.$$

For giving the complete description of $\mathcal{D}_3^0$ we define the sets

$$\begin{aligned}
S_1 := \{(x,y,z) \mid\ & 2x-2z \ge 1 \wedge 2x+2y+2z > -1 \wedge 2x+2y \le 1 \\
& \wedge 2x \le 1 \wedge 2x-2y+2z \le 1\},\\
S_2 := \{(x,y,z) \mid\ & x-z \le -1 \wedge 2x-2y+2z \le 1 \wedge -2x+2y \le 1 \\
& \wedge 2x > -1\},\\
S_3 := \{(x,y,z) \mid\ & x-z > -1 \wedge 2x-2y+2z \le 1 \wedge -2x+2y < 1, 2x > -1 \\
& \wedge 2x-2z < -1 \wedge 2x+2y+2z > -1\},\\
S_4 := \{(x,y,z) \mid\ & 2x-2y+2z \le 1 \wedge -2x+2y \le 1 \wedge 2x-2z = -1 \\
& \wedge 2x+2y+2z > -1\},\\
S_5 := \{(x,y,z) \mid\ & -1 < 2x \le 1 \wedge -1 < 2x-2z \le 1 \wedge 2x+2y+2z > -1 \\
& \wedge 2x-2y+2z \le 1 \wedge 2x+4y-2z < 3, 2y \le 1\}
\end{aligned}$$

and denote their union by

$$\mathcal{S} := \bigcup_{i\in\{1,\ldots,5\}} S_i.$$

Note that S_1, S_2, S_3, S_5 are polyhedra while S_4 is a polygon.

6.0.14 Theorem. $\mathcal{D}_3^0 = \mathcal{S}$

We give an outline of the proof. In a first step we will use Theorem 6. in order to show that

$$\mathcal{S} \subseteq \mathcal{D}_3^0. \tag{6.0.3}$$

For showing the opposite inclusion we need a set of nonzero periods Π such that for $\mathcal{P} := \bigcup_{\pi\in\Pi} P(\pi)$ we have

$$\mathcal{S} \cup \mathcal{P} \supseteq \mathcal{D}_3.$$

From (6.0.7) we can deduce $\mathcal{S} \cap \mathcal{P} = \emptyset$. Thus,

$$\mathcal{S} \supseteq \mathcal{D}_3 \setminus \mathcal{P} \supseteq \mathcal{D}_3^0.$$

Since $\mathcal{D}_3 \subset \overline{\mathcal{E}_3(1)}$ we are done if we can cover $\overline{\mathcal{E}_3(1)}$ with $\mathcal{P} \cup \mathcal{S}$, *i.e.*, if we can show that

$$\mathcal{P} \cup \mathcal{S} \supseteq \overline{\mathcal{E}_3(1)}.$$

There are 43 different periods, we denote the corresponding polyhedron by $P(\pi_j)$, where $j \in \{1, \ldots, 43\}$.

Cryptographically Secure Electronic Elections

In chapter four we detail all the protocol building blocks that we applied in our election schemes. In chapter five after describing requirements and participants of voting schemes two new secure election protocols are detailed. Both of them possess all basic requirements and can be implemented in practice.
Results of this chapter are based on [38] and [39].
Requirements we intend to fulfill in an electronic voting scheme are as follows: eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability, receipt-freeness, uncoercibility and protects against

randomization, forced-abstention and simulation attacks. A scheme is called *coercion-resistant* if it offers not only receipt-freeness, but also defense against randomization, forced-abstention and simulation attacks.

A Coercion-Resistant Voting Scheme Based on Blind Signatures

There are several election protocols using blind signatures that possess all basic requirements including verifiability, eligibility, unreusability, privacy etc., but not receipt-freeness ([28],[58]). Most of the receipt-free schemes in literature apply untappable channels or voting booths([59]), that are not practical. Our scheme satisfies besides eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability, coercion-resistance as well. The voting scheme based on blind signatures, requires only two authorities, practical and does not employ complex primitives like zero-knowledge proofs or threshold cryptosystems. It is offered to be employed in an environment, where authorities participating do not collude and the Voting Authority does not collaborate with adversaries.

Let denote P, Q large primes, where $Q|(P-1)$ and $g \in \mathbb{Z}_P^*$ of order Q. Let us define the candidate list as $C_1, C_2, \ldots, C_n$. The three functions applied in the scheme: *vote*, $ifeligible$ and $verify$ are as follows.

1. $vote(V_{ID}, SK_V, x, a, C_i) \rightarrowtail ballot$, where V_{ID} is the voter's identification number, SK_V is the voter's secret key, x, a are random parameters and C_i is the selected candidate. The form of the ballot is $(V_{ID}||r||y, V_{ID}||v)$, where $r = E_{SK_V}(g)$, $y \equiv g^{-x} \pmod P$ and $v \equiv y^a \cdot C_i \pmod P$ and $||$ is the notation of concatenation.

2. $ifeligible(PK_V, r) \rightarrowtail \{0,1\}$, where PK_V is the voter's public key, r is a received value. It returns 1 if $D_{PK_V}(r) = g$ and 0 if this congruence is not satisfied.

3. $verify(PK_V, z, s, y) \rightarrowtail \{0,1\}$ calculates if $PK_V^z \equiv g^s \cdot y \pmod P$ congruence holds. It outputs 1 if it is correct and 0 otherwise. This function verifies if s sent by the voter is calculated well and by the same voter who previously *voted* with value y and public key PK_V, where element z is randomly generated by the Voting Authority.

It consists of three distinctive stages: *Authorizing*, *Voting* and *Tallying*. Participants besides voters are Registry that is manages the Authorizing stage and the Tallying stage, as well, and Voting Authority that is responsible for the Voting stage. During the Authorizing stage the voter

authenticates himself and receives his credentials(SK_V,V_{ID}) and the ElGamal public key of the Voting Authority(PK_A). Voting Authority gets the voter roll containing the corresponding ElGamal public keys(V_{ID}, PK_V) and all system parameters are generated(P,Q,g). During the Voting stage voters create their ballots with function *vote*. Ballots contain the selected candidate and blind signature is applied to hide it from the Voting Authority (construction of value v). Voting Authority checks eligibility of the voters with function $ifeligible$ and if they have already voted before. Voting Authority sends an encrypted random number(z) to the voter. Voters send encrypted values s and V_{ID}, where $s \equiv x + z \cdot SK_V \pmod{Q}$, then Voting Authority runs function $verify$. Voters receive their encrypted ballots signed by the Voting Authority($Sig(v, s)$), if a fraud is detected the voter makes a claim. At the end voters pass the corresponding decrypting keys of the encrypted ballots (a,s) to the Registry. Ballots and bulletin board information are passed through an anonymous channel. During the Tallying stage the Voting Authority sends encrypted ballots (s,v) to the Registry. The ballots are being decrypted and the final results with the votes are listed on the bulletin board (s,C_i). Voters confirm that their ballots are on the bulletin board. If his ballot is not listed correctly, he makes a claim. During the voting process public and anonymous channels are employed and ElGamal encrypted messages are sent, hence it can be implemented in practice.

A Receipt-Free Homomorphic Election Scheme

Our protocol is based on homomorphic encryptions, it assumes existence of several authorities and it uses distributed ElGamal encryption [63]. This scheme is based on [22] that is not possessing the property of receipt-freeness or uncoercibility. There are two models based on [22] that are designed to be receipt-free in the literature: [52] and [36]. First one applies an honest verifier, the second one uses an untappable channel. Our scheme does not employ voting booths or untappable channels, it requires an anonymous return channel, hence it can be implemented in practice. We do not have an honest verifier, either. The only assumption is that among the Voting Authorities participating in distributed key generation and decryption there is at least one authority that is honest. The scheme satisfies eligibility, privacy, unreusability, fairness, robustness, individual and universal verifiability, receipt-freeness, uncoercibility and protects against randomization and forced-abstention attacks. The participants of the protocol are m voters, a Registry $\mathcal{R}$, an authority called Verifier Authority (*VA*) and

s Voting Authorities. Before describing our election scheme let us detail $ProofGenEG$ generator and $ProofVerEG$ verifier algorithms:

ProofGenEG

Input: signature: $s_m \in \mathbb{Z}_Q, R \in \mathbb{Z}_Q, \tilde{l} \in \mathbb{Z}_Q$

Output: $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$

1. The voter chooses random number: $\tilde{v} \in \mathbb{Z}_Q$
2. $R' \equiv (R \pmod P) \pmod Q$
3. $\overline{s_m} \equiv \frac{s_m}{\tilde{l}} \pmod Q$
4. $\overline{R} \equiv R^{\frac{\tilde{v}}{\tilde{l}}} \pmod P$
5. $\overline{T} \equiv \frac{R'}{\tilde{v}} \pmod Q$

ProofVerEG

Let denote EPK_{VA} Verifier Authority's ElGamal public key.

Input: $m \in \mathbb{Z}_P$, $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$

Output: true, false

1. $m' \equiv (m \pmod P) \pmod Q$
2. Verifies: $EPK_{VA}^{\overline{s_m}} \cdot \overline{R}^{\overline{T}} \equiv g^{m'} \pmod P$

During the Authorizing stage voters authenticate themselves in person and receive their credentials. All system parameters, sufficient private and public keys are generated. Let P and Q be large primes so that $Q|(P-1)$. G_Q denotes $\mathbb{Z}_P^*$'s unique multiplicative subgroup of order Q, and let g an arbitrary element such that $g \in G_Q$. Voting Authorities generate jointly the public (g, $h \equiv g^K \pmod P$) and private ($K \in \mathbb{Z}_Q$) keys using distributed ElGamal key generation method [63]. $\mathcal{R}$ randomly chooses $v_i \in \mathbb{Z}_Q^*$, $i = 1, \ldots, n$ elements $C_i \equiv g^{v_i} \pmod P$ where C_i represents candidate i from the voter roll and a one-way hash function $M()$ is chosen. All private and public keys are generated RSA keys of $\mathcal{R}$ (private: $RSK_{\mathcal{R}}, P_{\mathcal{R}}, Q_{\mathcal{R}}$, public: $RPK_{\mathcal{R}}, N_{\mathcal{R}}$) and *VA* (private: $RSK_{\mathcal{VA}}, P_{\mathcal{VA}}, Q_{\mathcal{VA}}$, public: $RPK_{\mathcal{VA}}, N_{\mathcal{VA}}$), ElGamal keys of *VA* (private: ESK_{VA}, public: (EPK_{VA}, P, g)). The voter gets his credential in a way that he generates his random reference number ($id_k^{\mathcal{R}}$), and $\mathcal{R}$ signs it blindly, hence $\mathcal{R}$ cannot connect the credential to the voter. During key-generation $\mathcal{R}$ does not learn anything about private keys either.

During the Voting stage voters create their ballots. *VA* checks eligibility of the voters and if they have already voted before by verifying signature of $\mathcal{R}$ on $id_k^{\mathcal{R}} \pmod{N_{\mathcal{R}}} \,||\, (M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \pmod{N_{\mathcal{R}}}$. Voter receives an

identification value used only in vote validation phase, in order to follow if a voter has already run the zero-knowledge proof. Voter V_k initiates a blind signature algorithm in order to get his identification number authorized and possesses $id_k^{VA} \pmod{N_{VA}}||(M(id_k^{VA}))^{RSK_{VA}} \pmod{N_{VA}}$. Then V_k sends $id_k^{VA} \pmod{N_{VA}}||(M(id_k^{VA}))^{RSK_{VA}} \pmod{N_{VA}}$ through an anonymous return channel to *VA*. *VA* verifies the signature and if the corresponding voter has not been processed before, sends z_k back through the same channel, where $z_k \in \mathbb{Z}_Q$ random. Since id_k^{VA} signed blindly and anonymous return channel is used, *VA* cannot learn the sender. V_k chooses a candidate i and the corresponding $C_i^{(k)}$ from $\mathcal{BB}$. In order to create his ballot randomly chooses $\alpha_k, \beta_k, \gamma_k \in \mathbb{Z}_Q$ and computes $G_k \equiv g^{\alpha_k+\beta_k} \pmod{P}$, $H_k \equiv h^{\alpha_k+\beta_k} \pmod{P}$ and $Y_k \equiv g^{z_k \cdot \gamma_k} \pmod{P}$. Following V_k runs a non-interactive zero-knowledge proof to prove that he has constructed the ballot correctly, such that he has chosen the value $C_i^{(k)}$ from the voter roll listed on $\mathcal{BB}$.
He chooses $r_j, d_j, w_k \in \mathbb{Z}_Q$ random numbers, where $1 \leq j \leq n$ and $j \neq i$, then calculates $(\mathsf{A}, \mathsf{B}) = (a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n)$, where $a_i \equiv g^{w_k} \pmod{P}$, $b_i \equiv h^{w_k} \pmod{P}$, for the elected candidate i and

$$a_j \equiv g^{r_j} \cdot G_k^{d_j} \pmod{P},\ b_j \equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}}\right)^{d_j} \pmod{P}$$

for all candidates $j \neq i$. Further, the voter calculates
$c_k = M(a_1||..||a_n||b_1||..||b_n||G_k||H_k \cdot C_i^{(k)}||g||h||id_k^{VA}||(M(id_k^{VA}))^{RSK_{VA}})$ challenge and $(\mathsf{D}, \mathsf{R}) = (d_1, r_1), (d_2, r_2), \ldots, (d_n, r_n)$ where for candidate i

$$d_i \equiv c_k - \sum_{j=1, i \neq j}^{n} d_j \pmod{Q},\ r_i \equiv w_k - (\alpha_k + \beta_k) \cdot d_i \pmod{Q}.$$

V_k sends the following encrypted randomized ballot and parameters to *VA* through an anonymous return channel:

$$(\mathsf{A}, \mathsf{B})||G_k||H_k \cdot C_i^{(k)}||c_k||(\mathsf{D}, \mathsf{R})||id_k^{VA}||(M(id_k^{VA}))^{RSK_{VA}}||\tilde{r} \cdot Y_k,$$

where $\tilde{r} \in \mathbb{Z}_P$ is random. After receiving all necessary information *VA* checks whether the voter with id_k^{VA} has already run the zero-knowledge

proof, whether id_k^{VA} is signed correctly and calculates the following congruences.

$$\begin{aligned} c_k &\equiv \sum_{j=1}^{n} d_j \pmod{Q}, \\ a_j &\equiv g^{r_j} \cdot G_k^{d_j} \pmod{P},\ j = 1, \ldots, n \\ b_j &\equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}} \right)^{d_j} \pmod{P},\ j = 1, \ldots, n \end{aligned}$$

If the verification congruences hold, then *VA* signs all the randomized components applying *SigGenEG* that is a Meta-ElGamal signature scheme [37]. *VA* calculates and sends $SigGenEG(G_k) = (s_{m_1}, R_1)$, $SigGenEG(H_k \cdot C_i^{(k)} \cdot Y_k \cdot \tilde{r}) = (s_{m_2}, R_2)$ $SigGenEG(Y_k \cdot \tilde{r}) = (s_{m_3}, R_3)$ back to the sender through the anonymous return channel. After the voter verifies the three signatures, gets authorization of the ballots being processed during the Tallying Stage.

$$\begin{aligned} \tilde{l_1} &\equiv (g^{\beta_k} \pmod{P}) \pmod{Q} \\ \tilde{l_2} &\equiv (h^{\beta_k} \cdot \tilde{r} \pmod{P}) \pmod{Q} \\ \tilde{l_3} &\equiv (\tilde{r} \pmod{P}) \pmod{Q} \end{aligned}$$

and computes

$$ProofGenEG(s_{m_1}, R_1, \tilde{l_1}) = (\overline{s_{m_1}}, \overline{R_1}, \overline{T_1})$$

$$ProofGenEG(s_{m_2}, R_2, \tilde{l_2}) = (\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$$

$$ProofGenEG(s_{m_3}, R_3, \tilde{l_3}) = (\overline{s_{m_3}}, \overline{R_3}, \overline{T_3}),$$

where *ProofGenEG* for generating a proof of his 'pure' ballots from the randomized ballot signatures sent by *VA*. Voters send $id_k^{\mathcal{R}} || g^{\alpha_k} || (\overline{s_{m_1}}, \overline{R_1}, \overline{T_1}) || h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k || (\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$ to $\mathcal{BB}$ through a public channel and $Y_k || (\overline{s_{m_3}}, \overline{R_3}, \overline{T_3})$ to *VA* through anonymous channel. The form of the ballot is the ElGamal encryption of $C_i^{(k)} \cdot Y_k \equiv g^{v_i + z_k \cdot \gamma_k} \pmod{P}$, where $z_k \in \mathbb{Z}_Q$ is sent by *VA* through an anonymous channel, hence z_k is not known by the adversary. If

the ballot appearing on $\mathcal{BB}$ is different or missing, then the voter makes a claim and he can cast his vote again.

During the Tallying stage the following computations are made: Verifier Authority runs *ProofVerEG* algorithm for each Y_k and calculates $Y \equiv \prod_{k=1}^{m} Y_k \pmod{P}$, where only valid randomized components are considered and sends Y to $\mathcal{BB}$. After verifying validity of encrypted ballots with *ProofVerEG*

$$\begin{aligned} \Gamma &\equiv \prod_{k=1}^{m} g^{\alpha_k} \pmod{P} \\ \Lambda &\equiv \prod_{k=1}^{m} h^{\alpha_k} \cdot C_i^{(k)} \cdot Y_k \pmod{P} \end{aligned}$$

appear on $\mathcal{BB}$, where only valid ballots are considered. After dividing Λ by Y we get the ElGamal encrypted voting result on $\mathcal{BB}$. Voting Authorities $A_1, A_2, \ldots, A_s$ together calculate the result $C_1^{t_1} \cdot C_2^{t_2} \cdots C_n^{t_n}$ with distributed ElGamal decryption method. Shanks baby step giant step or Pollard rho method might be applied for calculating t_i, $i = 1, \ldots, n$, which gives the election result for candidate i.

Calculation of $t_1, \ldots, t_n$ is considered as a computationally hard problem, it requires $O(m^{(n-1)/2})$ time to get the result.([52]) This scheme can be used for large scale election, if the authorities divide the total value of (Γ, Λ) into parts of reasonable size (e.g. election areas).

Összefoglaló

Ez a disszertáció két, többé-kevésbé független témakörön alapszik, az általánosított számrendszerek, illetve a biztonságos elektronikus választások témakörén. Az első részben kanonikus számrendszereket (CNS) vizsgálunk negyedfokú algebrai számtestekben, majd a háromdimenziós szimmetrikus shift radix rendszereket karakterizáljuk. A disszertáció második felében két biztonságos választási protokollt mutatunk be, az egyik vak aláírási technikán, a másik homomorf kriptorendszeren alapszik.

Általánosított számrendszerek

A második fejezet témája a **kanonikus számrendszerek** (CNS) karakterizálása. A CNS bázisokat explicite ismerjük néhány másodfokú, harmadfokú és negyedfokú testben (lásd a [43],[44],[30],[33],[86],[8],[49],[5],[67] dolgozatokat). Fő eredményként több algebrai számtestben, a negyedfokú körosztási, a legegyszerűbb negyedfokú testekben és a negyedfokú számtestek rendjeinek két családjában, meghatároztuk a CNS bázisokat. Ebben a fejezetben szereplő, Horst Brunotte-val és Pethővel Attilával közös eredményeink a [17] cikkünkben találhatók meg.

Továbbiakban $\mathbb{Q}$ jelöli a racionális számok testét, $\mathbb{Z}$ az egész számok halmazát és $\mathbb{N}$ a nemnegatív egészek halmazát. Jelölje $\mu_\gamma \in \mathbb{Z}[X]$ a γ algebrai egész minimálpolinomját és $\mathcal{C}_\gamma$ a $\mathbb{Z}[\gamma]$ összes CNS bázisának halmazát.

1. Definíció Jelölje $P(X) = X^d + p_{d-1}X^{d-1} + \cdots + p_1X + p_0 \in \mathbb{Z}[X]$, $N = \{0, 1, \ldots, |p_0| - 1\}$ és $\mathcal{R} := \mathbb{Z}[X]/P(X)\mathbb{Z}[X]$. A $\mathbb{Z}[X]$ -ből $\mathcal{R}$-be képező kanonikus epimorfizmus X -et vigye át x-be. Ha bármely nem-nulla $A(x) \in \mathcal{R}$ egyértelműen felírható $A(x) = a_0 + a_1x + \cdots + a_lx^l$ alakban, ahol $a_0, \ldots, a_l \in N, a_l \neq 0$, akkor (P, N) kanonikus számrendszer (CNS). $P(X)$-et CNS polinomnak, N-et számjegyek halmazának nevezzük.

Jelölje $\mathcal{C}$ a CNS polinomok halmazát. Belátható, hogy α akkor és csak akkor CNS bázis $\mathbb{Z}[\alpha]$-ban, ha μ_α CNS polinom. Hogy egy adott polinom CNS-e vagy sem, az algoritmus segítségével könnyen eldönthető.

B. Kovács [47] egyik tétele alapján egy rendben akkor és csak akkor létezik CNS, ha létezik hatvány egész bázis. CNS bázisok meghatározására B. Kovács és A. Pethő [49] algoritmusának egy módosított változatát alkalmazzuk. Az algoritmus ismertetéséhez, szükségünk van a következő állításokra és definícióra.

1. Lemma (B. Kovács – A. Pethő) Bármely nem-nulla α algebrai egész esetén a következő konstansok effektíve kiszámíthatóak:

$$k_\alpha = \min\{k \in \mathbb{Z} \,|\, \mu_\alpha(X+n) \in \mathcal{K} \text{ bármely } n \in \mathbb{Z}, \text{ ahol } n \geq k\},$$

$$c_\alpha = \min\{k \in \mathbb{Z} \,|\, \mu_\alpha(X+k) \in \mathcal{C}\}.$$

2. Definíció Az α algebrai egész *R alap CNS bázisa*, ha teljesül a következő két tulajdonság:

(1) $\alpha - n$ R CNS bázisa bármely $n \in \mathbb{N}$ esetén.

(2) $\alpha + 1$ nem CNS bázis R-ben.

1. Tétel Legyen γ egy algebrai egész. Akkor léteznek $\mathcal{F}_0(\gamma), \mathcal{F}_1(\gamma) \subset \mathcal{C}_\gamma$ véges, effektíve kiszámolható, diszjunkt halmazok, melyekre:

(i) Bármely $\alpha \in \mathcal{C}_\gamma$ esetén létezik olyan $n \in \mathbb{N}$, ahol $\alpha+n \in \mathcal{F}_0(\gamma) \cup \mathcal{F}_1(\gamma)$.

(ii) $\mathcal{F}_1(\gamma)$ elemei $\mathbb{Z}[\gamma]$ alap CNS bázisai.

Az algoritmus az 1. tételbeli (i) és (ii) tulajdonságokkal rendelkező $\mathcal{F}_0(\gamma)$ és $\mathcal{F}_1(\gamma)$ halmazokat adja meg.

Az algoritmus a következő:

Input: A γ nem-nulla algebrai egész és $\mathcal{B}$ (véges) halmaz, amely a $\mathbb{Z}[\gamma]$ hatvány egész bázisai ekvivalencia osztályainak reprezentásaiból áll.
Output: Az $\mathcal{F}_0(\gamma)$ és $\mathcal{F}_1(\gamma)$ halmazok.

1. [Inicializálás] Legyen $\{\beta_1, \ldots, \beta_t\} = \mathcal{B} \cup (-\mathcal{B})$, $F_0 = F_1 = T = \emptyset$ és $i = 1$.
2. [Minimál polinom kiszámítása] Legyen $P = \mu_{\beta_i}$.
3. [Van eleme az $F_0 \cup F_1$ halmaznak?] Ha létezik $k \in \mathbb{Z}, \delta \in \{0, 1\}$, hogy

$(P, k, \delta) \in T$, akkor tegye a $\beta_i - k$ értéket az F_δ halmazba és menjen a 11-es lépésre.
4. [Az alsó és felső határ meghatározása] Számítsa ki k_{β_i} és c_{β_i} értékeket.
5. [Elem beszúrása az F_1 halmazba] Ha $k_{\beta_i} - c_{\beta_i} \leq 1$, akkor szúrja be a $\beta_i - c_{\beta_i}$ értéket az F_1 halmazba, a $(P, c_{\beta_i}, 1)$-t a T-be és menjen a 11-es lépésre, egyébként menjen a 6-os lépésre az $l = c_{\beta_i} + 1, \ldots, k_{\beta_i} - 1$ értékkel, legyen $p_{k_{\beta_i}} = 1, k = c_{\beta_i}$ és lépjen a 8-as lépésre.
6. [CNS tulajdonság ellenőrzése] Ha $P(X + l) \in \mathcal{C}$, akkor legyen $p_l = 1$, egyébként $p_l = 0$.
7. [CNS bázis feltétel ellenőrzése] Ha $p_k = 0$, akkor lépjen a 9-es pontra.
8. [Elem $F_0 \cup F_1$ halmazba való beszúrása] Ha $p_{k+1} = \cdots = p_{k_{\beta_i}} = 1$, akkor szúrja be $\beta_i - k$ értéket az F_1 halmazba, $(P, k, 1)$-t T-be és lépjen a 11-es pontra, egyébként szúrja be $\beta_i - k$-t az F_0 halmazba és $(P, k, 0)$-t T-be.
9. [A k következő értéke] Legyen $k \leftarrow k + 1$.
10. [Befejeződött a CNS bázis ellenőrzése?] Ha $k \leq k_{\beta_i} - 1$, akkor menjen 7-re.
11. [Következő generátor] Legyen $i \leftarrow i + 1$.
12. [Vége?] Ha $i \leq t$, akkor menjen 2-re.
13. [Megáll] Az $\mathcal{F}_0(\gamma) = F_0$ és $\mathcal{F}_1(\gamma) = F_1$ halmazok listázása és az algoritmus befejeződése.

Térjünk át a 4-edfokú körosztási testekre.

2. Tétel Legyen $\zeta_5, \zeta_8, \zeta_{12}$ ötödik, nyolcadik és tizenkettedik primitív egységgyök. Ekkor $\mathcal{F}_0(\mathbb{Q}(\zeta_i)) = \emptyset$, ahol $i \in \{5, 8, 12\}$, továbbá
$\mathcal{F}_1(\mathbb{Q}(\zeta_5)) = \{-2 + \zeta_5, -3 - \zeta_5, -2 + \zeta_5 + \zeta_5^3, -3 - \zeta_5 - \zeta_5^3\}$,
$\mathcal{F}_1(\mathbb{Q}(\zeta_8)) = \{-3 \pm \zeta_8^k \mid k = 1, 3, 5, 7\}$,
$\mathcal{F}_1(\mathbb{Q}(\zeta_{12})) = \{-3 + \zeta_{12}, -3 - \zeta_{12}, -3 + \zeta_{12}^{-1}, -3 - \zeta_{12}^{-1}, -1 - \zeta_{12}^2 + \zeta_{12}^{-1}, -2 + \zeta_{12}^2 - \zeta_{12}^{-1}\}$.

Adott $t \in \mathbb{Z} \setminus \{0, \pm 3\}$ esetén jelölje $P_t(X)$ az $X^4 - tX^3 - 6X^2 + tX + 1$ polinomot. Legyen $\vartheta = \vartheta_t$ a $P_t(X)$ polinom egyik gyöke, ekkor a $K_t = K = \mathbb{Q}(\vartheta_t)$ számtestek végtelen parametrikus családját a *legegyszerűbb negyedfokú számtesteknek* nevezzük. P. Olajos [57] bebizonyította, hogy K_t akkor és csak akkor rendelkezik hatvány egész bázissal, ha $t = 2$ és $t = 4$, továbbá ezen testek hatvány egész bázisainak az összes generátorát meghatározta. Használva ezt az eredményt az összes CNS bázist ezekben a testekben.

3. Tétel $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$, $\mathcal{F}_1(\mathbb{Q}(\vartheta_2)) = \mathcal{G}_2$ és $\mathcal{F}_1(\mathbb{Q}(\vartheta_4)) = \mathcal{G}_4$, ahol $\mathcal{G}_2$ egy 19, míg $\mathcal{G}_4$ egy 12 elemű az értekezésben expliciten megadott halmaz.

K_t-beli $\mathbb{Z}[\alpha]$ polinomrend hatvány egész bázisait G. Lettl és A. Pethő [53] vizsgálta. Ezt alkalmazva bizonyítjuk be a következő tételt.

4. Tétel Legyen $t \in \mathbb{N} \setminus \{0, 3\}$ és jelölje ϑ az $X^4 - tX^3 - 6X^2 + tX + 1$ polinom egyik gyökét. Ekkor $\mathcal{F}_0(\mathbb{Q}(\vartheta)) = \emptyset$ és $\mathcal{F}_1(\mathbb{Q}(\vartheta)) = \mathcal{G} \cup \mathcal{G}_1 \cup \mathcal{G}_2 \cup \mathcal{G}_4$, ahol

$$\mathcal{G} = \begin{cases} \{-3-\vartheta, -t-2+\vartheta, -2-6\vartheta-t\vartheta^2+\vartheta^3, \\ -t-3+6\vartheta+t\vartheta^2-\vartheta^3\}, \text{ ha } t \geq 5, \\ \emptyset \text{ egyébként}, \end{cases}$$

$$\mathcal{G}_1 = \begin{cases} \{-4+\vartheta, -4-\vartheta, -5+6\vartheta+\vartheta^2-\vartheta^3, \\ -3-6\vartheta-\vartheta^2+\vartheta^3, -23+3\vartheta^2-\vartheta^3, -1-3\vartheta^2+\vartheta^3, \\ -14+25\vartheta+2\vartheta^2-4\vartheta^3, -10-25\vartheta-2\vartheta^2+4\vartheta^3\}, \\ \text{ ha } t = 1, \\ \emptyset \text{ egyébként}, \end{cases}$$

$$\mathcal{G}_2 = \begin{cases} \{-5+\vartheta, -3-\vartheta, -5+6\vartheta+2\vartheta^2-\vartheta^3, \\ -3-6\vartheta-2\vartheta^2+\vartheta^3\}, \text{ ha } t = 2, \\ \emptyset \text{ egyébként}, \end{cases}$$

$$\mathcal{G}_4 = \begin{cases} \{-6+\vartheta, -3-\vartheta, 1+9\vartheta-22\vartheta^2+4\vartheta^3, \\ -78-9\vartheta+22\vartheta^2-4\vartheta^3, -7+6\vartheta+4\vartheta^2-\vartheta^3, \\ -3-6\vartheta-4\vartheta^2+\vartheta^3, -62+74\vartheta+30\vartheta^2-9\vartheta^3, \\ -15-74\vartheta-30\vartheta^2+9\vartheta^3\}, \text{ ha } t = 4, \\ \emptyset \text{ egyébként}. \end{cases}$$

Tekintsük a paraméterezett negyedfokú számtestek rendjeinek egy családját, ahol az összes hatvány egész bázis ismert. Legyen $t \in \mathbb{Z}$, $t \geq 0$, és $P(X) = X^4 - tX^3 - X^2 + tX + 1$. Jelölje α a $P(X)$ polinom egyik gyökét. A következőkben $\mathcal{O} = Z[\alpha]$, $\mathbb{Q}(\alpha)$-beli rendet vizsgáljuk. M. Mignotte, A. Pethő és R. Roth [55] munkája alapján a következő eredményt kapjuk.

5. Tétel Legyen $t \geq 4$. Ekkor $\mathcal{F}_0(\mathbb{Q}(\alpha)) = \emptyset$ és $\mathcal{F}_1(\mathbb{Q}(\alpha)) = \mathcal{G}_4 \cup \mathcal{G}_t$, ahol

$$\begin{aligned}\mathcal{G}_4 &= \left\{209\alpha + 140\alpha^2 - 49\alpha^3 + 350, 209\alpha - 312\alpha^2 + 64\alpha^3 - 71\right\}\\ \mathcal{G}_t &= \left\{\alpha + t + 1, \alpha + t\alpha^2 - \alpha^3 + t + 2, t\alpha + (t-1)\alpha^2 - \alpha^3 + 8,\right.\\ &\quad t\alpha - (t+1)\alpha^2 + \alpha^3 + 2, \alpha - \alpha^3 + 2,\\ &\quad \left.\alpha - t(t^2+1)\alpha^2 + t^2\alpha^3 - t + 1\right\}.\end{aligned}$$

A harmadik fejezet témája a **szimmetrikus shift radix rendszerek**. Akiyama and Scheicher foglalkozott a kétdimenziós SSRS [9] rendszerekkel, mi a háromdimenziós SSRS esetet vizsgáljuk. Ez a fejezet a Claus Scheicher, Paul Surer és Jörg M. Thuswaldnerrel közös cikk [40] eredményeit taglalja.

3. Definíció ([9]) Legyen $d \geq 1$ egész, $\mathbf{r} \in \mathbb{R}^d$, és jelölje $\tau_{\mathbf{r}} : \mathbb{Z}^d \to \mathbb{Z}^d$, azt a leképezést, mely során az $\mathbf{a} = (a_1, \ldots, a_d)$ képe $\left(a_2, \ldots, a_d, -\left\lfloor \mathbf{ra} + \frac{1}{2} \right\rfloor\right)$. Ekkor $\tau_{\mathbf{r}}$-et *szimmetrikus shift radix rendszernek* (SSRS) hívjuk, ha $\forall \mathbf{a} \in \mathbb{Z}^d \quad \exists n \in \mathbb{N} : \tau_{\mathbf{r}}^n(\mathbf{a}) = \mathbf{0}$.

Legyen

$$\mathcal{D}_d := \left\{\mathbf{r} \in \mathbb{R}^d \,\middle|\, \forall \mathbf{a} \in \mathbb{Z}^d \, \exists n, l \in \mathbb{N} : \tau_{\mathbf{r}}^k(\mathbf{a}) = \tau_{\mathbf{r}}^{k+l}(\mathbf{a}) \ \forall k \geq n\right\} \text{ és}$$

$$\mathcal{D}_d^0 := \left\{\mathbf{r} \in \mathbb{R}^d \,|\tau_{\mathbf{r}} \text{ SSRS}\right\}.$$

A [9] cikkben szereplő algoritmus alapján bebizonyítjuk, hogy $\mathcal{D}_3^0$ négy test és egy sokszög egyesítése.

A [9] cikkben megmutatták, hogy

$$\mathcal{E}_d(1) \subset \mathcal{D}_d \subset \overline{\mathcal{E}_d(1)}. \tag{6.0.4}$$

Egy adott $\mathbf{r} = (r_1, \ldots, r_d) \in \mathcal{D}_d$ esetén, az $\mathbf{a} = (a_1, \ldots, a_d) \in \mathbb{Z}^d \setminus \{0\}$ elem az L periódushoz tartozó $\tau_{\mathbf{r}}$ egy *nem-nulla periódikus pontja*, ha $\mathbf{a} = \tau_{\mathbf{r}}^L(\mathbf{a})$. $\mathcal{D}_d^0$ definíciójából következik, hogy egy ilyen periódikus pont létezésének szükséges és elégséges feltétele, hogy $\mathbf{r} \notin \mathcal{D}_d^0$. Tegyük fel, hogy az $\mathbf{a}$ által definiált periódus átfut a

$$\tau_{\mathbf{r}}^j(\mathbf{a}) = (a_{1+j}, \ldots, a_{d+j}) \quad (0 \leq j \leq L-1)$$

íven, ahol $a_{L+1} = a_1, ..., a_{L+d-1} = a_{d-1}$. Jelöljön

$$(a_1, \ldots, a_d); a_{d+1}, \ldots, a_L$$

egy ilyen periódust, és ezt $\tau_{\mathbf{r}}$ periódusának, vagy egyszerűen $\mathcal{D}_d$ *periódusának* nevezzük.

Legyen $\pi := (a_1, \ldots, a_d); a_{d+1}, \ldots, a_L$ egy nem-nulla periódikus pont. Keressük azon $\mathbf{r} \in \mathcal{D}_d$ pontok $P(\pi)$ halmazát, amelyeknél π a $\tau_{\mathbf{r}}$ egy periódusaként áll elő. A $\tau_{\mathbf{r}}$ definíciója alapján, az $\mathbf{r} \in P(\pi)$ elem kielégíti a következő kétoldali egyenlőtlenség rendszert:

$$-\frac{1}{2} \leq r_1 a_{1+i} + r_2 a_{2+i} + \cdots + r_d a_{d+i} + a_{d+1+i} < \frac{1}{2}, \tag{6.0.5}$$

ahol i 0-tól $L-1$-ig megy és $a_{L+1} = a_1, \ldots, a_{L+d} = a_d$. Az ilyen rendszer egy konvex testet határoz meg, mely esetleg elfajuló, sőt üres is lehet. Ezért $P(\pi)$-t *kivágó testnek* nevezzük. Mivel az összes $\mathbf{r} \in P(\pi)$ pont rendelkezik a megfelelő $\tau_{\mathbf{r}}$ leképezés π periódusával a $P(\pi)$ halmaz és a $\mathcal{D}_d^0$ halmaz metszete üres. Így

$$\mathcal{D}_d^0 = \mathcal{D}_d \setminus \bigcup_{\pi \neq \mathbf{0}} P(\pi),$$

ahol az unió az összes nem-nulla π periódusra vonatkozik. Mivel a periódusok halmaza végtelen, ez a kifejezés nem alkalmas kalkulációkra. A következő tétel megmutatja, hogyan lehet lecsökkenteni az összes lehetséges periódusok halmazát véges halmazra, és megad egy hatékony algoritmust a $H \cap \mathcal{D}_d^0$ kiszámítására, ahol H egy zárt részhalmaza $\operatorname{int} \mathcal{D}_d = \mathcal{E}_d(1)$-nek. Legyen $\mathbf{e}_i$ az i-dik kanonikus egységvektor. Az $\mathbf{r} = (r_1, \ldots, r_d) \in \operatorname{int} \mathcal{D}_d$ esetén, jelölje $\mathcal{V}(\mathbf{r}) \subset \mathbb{Z}^d$ a legkisebb halmazt, mely a következő tulajdonságokkal rendelkezik:

1. $\pm\mathbf{e}_i \in \mathcal{V}(\mathbf{r}), i = 1, \ldots, d,$
2. $(a_1, \ldots, a_d) \in \mathcal{V}(\mathbf{r}) \Rightarrow (a_2, \ldots, a_{d+1}) \in \mathcal{V}(\mathbf{r})$ ahol a_{d+1} kielégíti a
$$-1 < r_1 a_1 + r_2 a_2 + \cdots + r_d a_d + a_{d+1} < 1.$$

$\mathcal{V}(\mathbf{r}) \subset \mathbb{Z}^d$ az $\mathbf{r}$ *tanúhalmazának* nevezzük. Ezen kívül $\mathcal{G}(\mathcal{V}(\mathbf{r})) = V \times E$ jelöljön egy gráfot, melynek csúcsainak halmaza $V = \mathcal{V}(\mathbf{r})$ és éleinek halmaza pedig $E \subset V \times V$ úgy, hogy

$$\forall \mathbf{a} \in V : (\mathbf{a}, \tau_{\mathbf{r}}(\mathbf{a})) \in E.$$

6. Tétel ([9]) Legyen $\mathbf{r}_1, \ldots, \mathbf{r}_k \in \mathcal{D}_d$ és legyen $H := \square(\mathbf{r}_1, \ldots, \mathbf{r}_k)$ az $\mathbf{r}_1, \ldots, \mathbf{r}_k$ pontok konvex burka. Tegyük fel, hogy $H \subset \operatorname{int} \mathcal{D}_d$ és mérete megfelelően kicsi. Akkor létezik egy olyan algoritmus, mely megad egy véges, irányított, $G(H) = V \times E$ gráfot, ahol a csúcsok halmaza $V \subset \mathbb{Z}^d$ és az élek halmaza $E \subset V \times V$, melyekre teljesül

1. $\pm\mathbf{e}_i \in V$, bármely $i = 1, \ldots, d$,
2. $\mathcal{G}(\mathcal{V}(\mathbf{x}))$ részgráfja $G(H)$-nek, bármely $\mathbf{x} \in H$,
3. $H \cap \mathcal{D}_d^0 = H \setminus \bigcup_\pi P(\pi)$, ahol π végigfut a G gráf nem-nulla egyszerű körei által indukált periódusokon.

Célunk a $\mathcal{D}_3^0$ karakterizálása. Már tudjuk, hogy

$$\mathcal{E}_3(1) \subset \mathcal{D}_3 \subset \overline{\mathcal{E}_3(1)},$$

továbbá a [78, 84] dolgozatok alapján

$$\mathcal{E}_3(1) = \{(x, y, z) \in \mathbb{R}^3 |\ |x| < 1, |y - xz| < 1 - x^2, |x + z| < |y + 1|\}.$$

adódik. Szükségünk van a $\overline{\mathcal{E}_3(1)}$ halmazra. Könnyen látható, hogy ha a szigorú egyenlőtlenségeket kicseréljük megengedőekre, még nem kapunk zárt halmazt. Meg kell adni még további egyenlőtlenségeket. Legyen

$$\begin{aligned}\mathcal{E}_3' := \big\{(x, y, z) \in \mathbb{R}^3 \big|\ |x| \le 1 \wedge |y - xz| \le 1 - x^2 \\ \wedge |x + z| \le |y + 1| \wedge |y - 1| \le 2 \wedge |z| \le 3\big\}\end{aligned} \tag{6.0.6}$$

és tekintsük az $\mathcal{E}_3'$ és az

$$A_c := \big\{(x, y, z) \in \mathbb{R}^3 \mid x - c = 0\big\}$$

sík metszetét egy adott c konstans esetén. A következő lemma megmutatja, hogy $\mathcal{E}_3'$ zárt.

2. Lemma Bármely $|c| < 1$ esetén az $\mathcal{E}_3'$ és az A_c sík metszete egy $\triangle(A_c^{(1)}, A_c^{(2)}, A_c^{(3)})$ zárt háromszög, ahol $A_c^{(1)} = (c, -1, -c), A_c^{(2)} = (c, 1 - 2c, c - 2), A_c^{(3)} = (c, 2c + 1, c + 2)$.

7. Tétel $\overline{\mathcal{E}_3(1)} = \mathcal{E}'_3$.

Az $\mathcal{E}'_3$ halmazt definiáló egyenlőtlenségek száma lecsökkenthető, a következőket kapjuk eredményül:

$$\overline{\mathcal{E}_3(1)} = \left\{(x,y,z) \,\middle|\, |x+z| \le 1+y \wedge y - xz \le 1 - x^2 \wedge |z| \le 3\right\}.$$

Ahhoz, hogy megadjuk $\mathcal{D}_3^0$ teljes karakterizációját, definiáljuk a következő halmazokat:

$$\begin{aligned}
S_1 := \{(x,y,z) \mid\ & 2x - 2z \ge 1 \wedge 2x + 2y + 2z > -1 \wedge 2x + 2y \le 1 \\
& \wedge 2x \le 1 \wedge 2x - 2y + 2z \le 1\},\\
S_2 := \{(x,y,z) \mid\ & x - z \le -1 \wedge 2x - 2y + 2z \le 1 \wedge -2x + 2y \le 1 \\
& \wedge 2x > -1\},\\
S_3 := \{(x,y,z) \mid\ & x - z > -1 \wedge 2x - 2y + 2z \le 1 \wedge -2x + 2y < 1 \\
& \wedge 2x > -1 \wedge 2x - 2z < -1 \wedge 2x + 2y + 2z > -1\},\\
S_4 := \{(x,y,z) \mid\ & 2x - 2y + 2z \le 1 \wedge -2x + 2y \le 1 \\
& \wedge 2x - 2z = -1, \wedge 2x + 2y + 2z > -1\},\\
S_5 := \{(x,y,z) \mid\ & -1 < 2x \le 1 \wedge -1 < 2x - 2z \le 1 \\
& \wedge 2x + 2y + 2z > -1 \wedge 2x - 2y + 2z \le 1 \\
& \wedge 2x + 4y - 2z < 3 \wedge 2y \le 1\}
\end{aligned}$$

és jelölje

$$\mathcal{S} := \bigcup_{i \in \{1,\ldots,5\}} S_i.$$

az egyesítésüket. Megjegyezzük, hogy S_1, S_2, S_3, S_5 testek, míg S_4 sokszög. A fenti jelölésekkel adódik a következő

8. Tétel $\mathcal{D}_3^0 = \mathcal{S}$.

A bizonyítás váza a következő. Első lépésként a 6. Tétel alapján belátjuk, hogy

$$\mathcal{S} \subseteq \mathcal{D}_3^0. \tag{6.0.7}$$

Ahhoz, hogy a másik irányú tartalmazást belássuk, szükségünk van a nemnulla periódusok Π halmazára. Legyen $\mathcal{P} := \bigcup_{\pi\in\Pi} P(\pi)$, továbbá be kell látnunk, hogy

$$\mathcal{S} \cup \mathcal{P} \supseteq \mathcal{D}_3.$$

A (6.0.7) reláció alapján $\mathcal{S} \cap \mathcal{P} = \emptyset$. Ebből következik, hogy

$$\mathcal{S} \supseteq \mathcal{D}_3 \setminus \mathcal{P} \supseteq \mathcal{D}_3^0,$$

azaz $\mathcal{D}_3^0 \subseteq \mathcal{S}$. Mivel $\mathcal{D}_3 \subset \overline{\mathcal{E}_3(1)}$, készen vagyunk, ha le tudjuk fedni $\overline{\mathcal{E}_3(1)}$-t a $\mathcal{P} \cup \mathcal{S}$ halmazzal. Számításokkal ez könnyen megmutatható.

Biztonságos elektronikus választások

A negyedik fejezet a választási protokollokban alkalmazott kriptográfiai primitíveket mutatja be. Az ötödik fejezetben, miután felsoroltuk a választási sémákkal szembeni elvárásokat, illetve a résztvevőket, két új, biztonságos szavazó protokollt ismertetünk. Mindkét protokoll rendelkezik a szükséges alapvető elvárásokkal és a gyakorlatban is implementálható. Ennek a fejezetnek az eredményei megtalálhatóak a [38] és [39] cikkekben.

Az elektronikus szavazó sémák elvárásai a következőek: jogosultság, titkosság, egyszer-szavazhatóság, szabályosság, teljesség, individuális és univerzális ellenőrizhetőség, visszaigazolás-mentesség. Ha egy protokoll viszszaigazolás-mentes, akkor a szavazó nem vesztegethető, illetve nem fenyegethető meg.

Egy protokoll *ellenálló*, ha visszaigazolás-mentes, és biztosított a véletlen-érték támadás, a kényszerített-hiányzás és a szimulációs támadásokkal szemben.

A fejezet első felében egy **vak aláíráson alapuló, ellenálló szavazó sémát** ismertetünk. Több olyan vak aláírási technikát alkalmazó választási protokoll is ismert, mely rendelkezik az alapvető elvárásokkal, mint például az ellenőrizhetőség, jogosultság, egyszer-szavazhatóság, titkosság stb., de nem visszaigazolás-mentes (lásd pl. a [28] és [58] dolgozatokat). Az irodalomban a legtöbb visszaigazolás-mentes séma lehallgathatatlan csatornát vagy szavazó fülke csatornát használ, ami nem gyakorlatias. A mi sémánk megfelel a jogosultság, titkosság, egyszer-szavazhatóság, szabályosság, teljesség, individuális és univerzális ellenőrizhetőség elvárásoknak, és ellenálló

is. A vak aláírási technikán alapuló sémánk két szervezet részvételét tételezi fel, gyakorlatias és nem tartalmaz bonyolult primitíveket, mint például nulla-ismeretű bizonyításokat vagy osztott kriptorendszereket. Ajánlott olyan környezetben implementálni, ahol a résztvevő szervezetek nem fognak össze és a Szavazó Bizottság nem működik együtt a támadókkal.

Jelöljön P, Q két nagy prímet, ahol $Q|(P-1)$ és legyen $g \in \mathbb{Z}_P^*$, melynek rendje Q. A jelöltek listája legyen $C_1, C_2, \ldots, C_n$. Három függvényt alkalmazunk: *vote*, *ifeligible* és *verify*.

1. $vote(V_{ID}, SK_V, x, a, C_i) \rightarrowtail ballot$, ahol V_{ID} a szavazó azonosító száma, SK_V a szavazó titkos kucsa, x, a véletlen paraméterek és C_i a javasolt jelölt. A *ballot* formátuma: $(V_{ID}||r||y, V_{ID}||v)$, ahol

$$\begin{aligned} r &= E_{SK_V}(g) \\ y &\equiv g^{-x} \pmod{P} \\ v &\equiv y^a \cdot C_i \pmod{P} \end{aligned}$$

   és $||$ a konkatenáció jele.

2. $ifeligible(PK_V, r) \rightarrowtail \{0,1\}$, ahol PK_V a szavazó nyilvános kulcsa, r input érték. A függvény 1-et ad vissza, ha $D_{PK_V}(r) = g$ és 0-t, ha a kongruencia nem teljesül.

3. $verify(PK_V, z, s, y) \rightarrowtail \{0,1\}$ kiszámolja, hogy a $PK_V^z \equiv g^s \cdot y \pmod{P}$ kongruencia teljesül-e. Ha teljesül 1-et ad vissza, ha nem, akkor 0-t. Ez a függvény azt ellenőrzi, hogy s-et szabályosan számították-e ki illetve, hogy ugyanaz a személy küldte-e az s értéket, aki megelőzően *szavazott* az y értékkel és a PK_V publikus kulccsal, a z a Szavazó Bizottság által generált véletlen szám.

A protokoll három jól elhatárolható fázisból áll: *Regisztáció*, *Szavazás* és *Összeszámlálás*. A szavazókon kívül résztvevők még a Hitelesítő Szervezet, mely a regisztrációt és az összeszámlálást vezényli, valamint a Szavazó Bizottság, mely a szavazó fázisért felelős.

A regisztráció során megtörténik a szavazó azonosítása, megkapja elektronikus azonosítóját (SK_V,V_{ID}), valamint a Szavazó Bizottság ElGamal nyilvános kulcsát (PK_A). A Szavazó Bizottság megkapja a szavazó

listát, mely tartalmazza a szavazásra jogosultak azonosítóját és nyilvános kulcsát (V_{ID}, PK_V), valamint megtörténik a szükséges rendszerparaméterek meghatározása (P,Q,g).

A szavazó fázis során a szavazók elkészítik az elektronikus szavazócédulájukat (*ballot*) a *vote* függvény segítségével. A szavazat tartalmazza a javasolt jelölt azonosítóját, vak aláírási technikával a Szavazó Bizottság hitelesíti azt (v konstrukciója). A Szavazó Bizottság ellenőrzi a szavazó jogosultságát az *ifeligible* függvény segítségével és megnézi szavazott-e már. A Szavazó Bizottság elküld egy titkosított z véletlen számot a szavazónak. A szavazó visszaküldi az s és V_{ID} értékeket, ahol $s \equiv x + z \cdot SK_V \pmod{Q}$, majd a Szavazó Bizottság lefuttatja a *verify* függvényt. A szavazásra jogosultak megkapják a hitelesített $Sig(v, s)$ elektronikus szavazatukat a Szavazó Bizottságtól, ha nem érvényes a szavazat, akkor a szavazó reklamál. A szavazó fázis lezárása után a szavazók elküldik a megfelelő a,s dekódoló kulcsot a Hitelesítő Szervezetnek. A szavazatok, illetve a hirdető táblára küldött információk anonim csatornán továbbítódnak.

Az összeszámlálás során a Szavazó Bizottság a titkosított s,v szavazatokat elküldi a Hitelesítő Szervezetnek. A szavazatokat dekódolják és a végleges eredménnyel együtt nyilvánosságra hozzák a hirdető táblán (s,C_i). A szavazók ellenőrzik, hogy a szavazatuk a táblán van-e. Ha a szavazatuk nem szerepel, vagy hibásan szerepel, akkor reklamálnak. Az egész szavazó eljárás alatt nyilvános és anonim csatornát alkalmazunk, valamint ElGamallal titkosított üzenetet továbbítottunk, így a rendszer gyakorlatias.

A fejezet második felében egy **visszaigazolás-mentes homomorf választási sémát** mutatunk be. A protokollunk homomorf titkosításon alapszik, több szervezet közreműködésével osztott ElGamal kriptorendszert használ (lásd [63]). Ennek a sémának az alapja a [22] dolgozatban szereplő protokoll, ami nem visszaigazolás-mentes. Két visszaigazolásmentes változatot is találunk az irodalomban, a [52] és a [36] cikkekben levő sémák. Az első egy teljesen megbízható ellenőrző szervezet részvételét tételezi fel, a másik lehallgathatatlan csatornát használ. A mi változatunk nem a szavazó fülke vagy lehallgathatatlan csatornát, hanem a gyakorlatias, anonim válasz csatornát alkalmazza. Nem tételezzük fel egyik szervezetről sem, hogy teljesen megbízható, az egyetlen feltételezés az, hogy a Szavazó Bizottságok között az osztott kulcsgenerálás és dekódolás során legalább egy megbízható. A séma megfelel az alapvető elvárásoknak: jogosultság, titkosság, egyszer-szavazhatóság, szabályosság, teljesség, individuális és

univerzális ellenőrizhetőség, visszaigazolás-mentesség és ellenáll a véletlenérték és kényszerített-hiányzás támadásoknak. A protokoll résztvevői az m szavazón kívül, az $\mathcal{R}$ Hitelesítő Szervezet, egy speciális szervezet, az Ellenőrző Szervezet (*VA*) és s Szavazó Bizottság.

Mielőtt rátérnénk a séma részletezésére megadjuk a $ProofGenEG$ generátor és $ProofVerEG$ ellenőrző algoritmust:

ProofGenEG
Input: aláírás: $s_m \in \mathbb{Z}_Q, R \in \mathbb{Z}_Q, \tilde{l} \in \mathbb{Z}_Q$
Output: $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$
1. A szavazó választ egy véletlen számot: $\tilde{v} \in \mathbb{Z}_Q$
2. $R' \equiv (R \pmod P) \pmod Q$
3. $\overline{s_m} \equiv \frac{s_m}{\tilde{l}} \pmod Q$
4. $\overline{R} \equiv R^{\frac{\tilde{v}}{\tilde{l}}} \pmod P$
5. $\overline{T} \equiv \frac{R'}{\tilde{v}} \pmod Q$

ProofVerEG
Jelölje EPK_{VA} az Ellenőrző Szervezet ElGamal nyilvános kulcsát.
Input: $m \in \mathbb{Z}_P$, $\overline{s_m} \in \mathbb{Z}_Q, \overline{R} \in \mathbb{Z}_P, \overline{T} \in \mathbb{Z}_Q$
Output: igaz, hamis
1. $m' \equiv (m \pmod P) \pmod Q$
2. Ellenőrzés: $EPK_{VA}^{\overline{s_m}} \cdot \overline{R}^{\overline{T}} \equiv g^{m'} \pmod P$

A regisztrációs fázisban a szavazók személyesen igazolják személyazonosságukat és megkapják elektronikus azonosítójukat. A szükséges rendszer-paraméterek, titkos és nyilvános kulcsok legenerálódnak. Legyen P és Q két nagy prím, ahol $Q|(P-1)$. G_Q jelölje $\mathbb{Z}_P^*$ multiplikatív részcsoportját, melynek rendje Q, és legyen $g \in G_Q$ egy tetszőleges elem. A Szavazó Bizottságok együttesen legenerálják a szükséges nyilvános (g, $h \equiv g^K \pmod P$) és titkos ($K \in \mathbb{Z}_Q$) kulcsokat osztott ElGamal kulcsgeneráló módszerrel [63]. $\mathcal{R}$ véletlenül választ $v_i \in \mathbb{Z}_Q^*$, $i = 1, \ldots, n$ elemeket, $C_i \equiv g^{v_i} \pmod P$, ahol C_i jelöli az i-edik jelöltet és egy $M()$ egyirányú hash függvényt. Az összes titkos és nyilvános kulcsot legenerálják: $\mathcal{R}$ RSA kulcsa (titkos: $RSK_{\mathcal{R}}, P_{\mathcal{R}}, Q_{\mathcal{R}}$, nyilvános: $RPK_{\mathcal{R}}, N_{\mathcal{R}}$) és *VA* RSA kulcsa (titkos: $RSK_{\mathcal{VA}}, P_{\mathcal{VA}}, Q_{\mathcal{VA}}$, nyilvános: $RPK_{\mathcal{VA}}, N_{\mathcal{VA}}$), *VA* ElGamal kulcsa (titkos: ESK_{VA}, nyilvános: (EPK_{VA}, P, g)). A szavazó úgy

kapja meg azonosítóit, hogy generál egy véletlen $id_k^{\mathcal{R}}$ referencia számot, és $\mathcal{R}$ vakon aláírja, így $\mathcal{R}$ nem tudja az azonosítót hozzárendelni magához a szavazóhoz. Természetesen a kulcsgenerálás során $\mathcal{R}$-nek nincs semmilyen információja a titkos kulcsokról sem.

A szavazó fázis során a szavazók elkészítik az elektronikus szavazatukat. *VA* ellenőrzi a szavazók jogosultságát, és hogy szavaztak-e már úgy, hogy ellenőrzik $\mathcal{R}$ aláírásának érvényességét, megvizsgálva az $id_k^{\mathcal{R}} \pmod{N_{\mathcal{R}}}$ és az $(M(id_k^{\mathcal{R}}))^{RSK_{\mathcal{R}}} \pmod{N_{\mathcal{R}}}$ értékeket. A szavazó kap egy azonosítót, mely csak a szavazat-ellenőrző fázisban szükséges, abból a célból, hogy ellenőrizzük, hogy a nulla-ismeretű bizonyítást lefuttatta-e. A V_k szavazó vak aláírást kezdeményez, hogy az azonosítóit hitelesítsék: $id_k^{VA} \pmod{N_{VA}}||(M(id_k^{VA}))^{RSK_{VA}} \pmod{N_{VA}}$. Majd V_k elküldi az $id_k^{VA} \pmod{N_{VA}}$ és $(M(id_k^{VA}))^{RSK_{VA}} \pmod{N_{VA}}$ üzeneteket egy anonim-válasz csatornán *VA*-nak. *VA* ellenőzi az aláírást, és ha a szavazóval még nem találkozott korábban, visszaküldi a $z_k \in \mathbb{Z}_Q$ véletlen értéket ugyanazon a csatornán. Mivel az id_k^{VA}-t vakon írták alá és anonim-válasz csatornát használnak, *VA* nem tudja a szavazó személyét. V_k kiválasztja az i-ik jelöltet és a megfelelő $C_i^{(k)}$ értéket a $\mathcal{BB}$-ről. Ahhoz, hogy elkészítse az elektronikus szavazatát választ $\alpha_k, \beta_k, \gamma_k \in \mathbb{Z}_Q$ véletlen számokat és kiszámolja a $G_k \equiv g^{\alpha_k+\beta_k} \pmod{P}$, $H_k \equiv h^{\alpha_k+\beta_k} \pmod{P}$ és $Y_k \equiv g^{z_k \cdot \gamma_k} \pmod{P}$ értékeket. V_k lefuttat egy nem-interaktív nulla-ismeretű bizonyítást, hogy bebizonyítsa az elektronikus szavazat szabályosságát, azaz, hogy a $C_i^{(k)}$ érték tényleg a jelöltek listájából vett. A szavazó választ $r_j, d_j, w_k \in \mathbb{Z}_Q$ véletlen számokat, ahol $1 \leq j \leq n$ és $j \neq i$, majd kiszámolja az $(\mathsf{A}, \mathsf{B}) = (a_1, b_1), (a_2, b_2), \cdots, (a_n, b_n)$ párokat, ahol

$$a_i \equiv g^{w_k} \pmod{P},$$
$$b_i \equiv h^{w_k} \pmod{P},$$

teljesül a kiválasztott i-edik jelöltre és

$$\begin{aligned} a_j &\equiv g^{r_j} \cdot G_k^{d_j} \pmod{P}, \\ b_j &\equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}} \right)^{d_j} \pmod{P} \end{aligned}$$

az összes többi j-edik jelöltre, $j \neq i$. Továbbá, a szavazó kiszámolja a

$$c_k = M(a_1||..||a_n||b_1||..||b_n||G_k||H_k \cdot C_i^{(k)}||g||h||id_k^{VA}||(M(id_k^{VA}))^{RSK_{VA}})$$

kihívást és a $(\mathsf{D},\mathsf{R}) = (d_1, r_1), (d_2, r_2), \ldots, (d_n, r_n)$ párokat
ahol az i-ik jelöltre

$$\begin{aligned} d_i &\equiv c_k - \sum_{j=1, i\neq j}^{n} d_j \pmod{Q} \\ r_i &\equiv w_k - (\alpha_k + \beta_k) \cdot d_i \pmod{Q} \end{aligned}$$

teljesül. V_k elküldi a következő titkosított randomizált szavazatot és paramétereket *VA*-nak anonim-válasz csatornát használva:

$$(\mathsf{A},\mathsf{B})||G_k||H_k \cdot C_i^{(k)}||c_k||(\mathsf{D},\mathsf{R})||id_k^{VA}||(M(id_k^{VA}))^{RSK_{VA}}||\tilde{r} \cdot Y_k,$$

ahol $\tilde{r} \in \mathbb{Z}_P$ véletlen szám. Miután megkapta az összes szükséges adatot, *VA* ellenőrzi, hogy a szavazó az id_k^{VA} azonosítóval lefuttatta-e már a nulla-ismeretű bizonyítást, hogy id_k^{VA} aláírása érvényes-e, és kiszámolja a következő kongruenciákat:

$$\begin{aligned} c_k &\equiv \sum_{j=1}^{n} d_j \pmod{Q}, \\ a_j &\equiv g^{r_j} \cdot G_k^{d_j} \pmod{P},\ j = 1, \ldots, n \\ b_j &\equiv h^{r_j} \cdot \left(\frac{H_k \cdot C_i^{(k)}}{C_j^{(k)}} \right)^{d_j} \pmod{P},\ j = 1, \ldots, n. \end{aligned}$$

Ha az ellenőrző kongruenciák teljesülnek, akkor *VA* aláírja az összes randomizált komponenst *SigGenEG* segítségével, ami egy Meta-ElGamal aláírási séma (lásd a [37] dolgozatot). *VA* kiszámolja és visszaküldi anonim-válasz csatornán a következő mennyiségeket a küldőnek:

$$\begin{aligned} SigGenEG(G_k) &= (s_{m_1}, R_1) \\ SigGenEG(H_k \cdot C_i^{(k)} \cdot Y_k \cdot \tilde{r}) &= (s_{m_2}, R_2) \\ SigGenEG(Y_k \cdot \tilde{r}) &= (s_{m_3}, R_3) \end{aligned}$$

Miután a szavazó ellenőrzi mindhárom aláírást, generálja a hitelesített

szavazatokat:

$$\begin{aligned}\tilde{l_1} &\equiv (g^{\beta_k} \pmod P) \pmod Q\\ \tilde{l_2} &\equiv (h^{\beta_k}\cdot\tilde{r} \pmod P) \pmod Q\\ \tilde{l_3} &\equiv (\tilde{r} \pmod P) \pmod Q\end{aligned}$$

és kiszámolja

$$ProofGenEG(s_{m_1}, R_1, \tilde{l_1}) = (\overline{s_{m_1}}, \overline{R_1}, \overline{T_1})$$

$$ProofGenEG(s_{m_2}, R_2, \tilde{l_2}) = (\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$$

$$ProofGenEG(s_{m_3}, R_3, \tilde{l_3}) = (\overline{s_{m_3}}, \overline{R_3}, \overline{T_3}),$$

ahol *ProofGenEG* generál egy bizonyítékot, mely biztosítja a 'tényleges' szavazatok érvényességét. A szavazó elküldi nyilvános csatornán $\mathcal{BB}$-re az $id_k^{\mathcal{R}}||g^{\alpha_k}||(\overline{s_{m_1}}, \overline{R_1}, \overline{T_1})||h^{\alpha_k}\cdot C_i^{(k)}\cdot Y_k||(\overline{s_{m_2}}, \overline{R_2}, \overline{T_2})$ üzenetet, és anonim csatornán az $Y_k||(\overline{s_{m_3}}, \overline{R_3}, \overline{T_3})$ értékeket továbbítja *VA*-nak. A szavazat az ElGamallal kódolt $C_i^{(k)}\cdot Y_k \equiv g^{v_i+z_k\cdot\gamma_k} \pmod P$ érték, ahol a $\mathbb{Z}_Q$-beli z_k értéket *VA* küldte anonim csatornán, így z_k a támadó számára ismeretlen. Ha a $\mathcal{BB}$-n levő szavazat különbözik, vagy hiányzik, akkor a szavazó reklamál és újra szavazhat.

Az összeszámlálási fázis során a következő számítások történnek: Az Ellenőrző Szervezet lefuttatja a *ProofVerEG* algoritmust minden egyes Y_k-ra és kiszámolja az $Y \equiv \prod_{k=1}^{m} Y_k \pmod P$ értéket, ahol csak az érvényes randomizált komponenseket veszi figyelembe, és elküldi Y-t a $\mathcal{BB}$-re. Miután ellenőrizte a titkosított szavazatok érvényességét a *ProofVerEG* algoritmussal, a

$$\begin{aligned}\Gamma &\equiv \prod_{k=1}^{m} g^{\alpha_k} \pmod P\\ \Lambda &\equiv \prod_{k=1}^{m} h^{\alpha_k}\cdot C_i^{(k)}\cdot Y_k \pmod P\end{aligned}$$

értékek megjelennek $\mathcal{BB}$-én, ahol csak az érvényes szavazatokat veszik figyelembe. Elosztva Λ-t Y-nal, a szavazás eredményének ElGamallal

titkosított képe lesz $\mathcal{BB}$-n. Az $A_1, A_2, \ldots, A_s$ Szavazó Bizottságok osztott ElGamal dekódolással együttesen kiszámolják a $C_1^{t_1} \cdot C_2^{t_2} \cdots C_n^{t_n}$ mennyiséget. Shanks baby step giant step vagy Pollard rho algoritmus alkalmazq-ható t_i, $i = 1, \ldots, n$ kiszámítására, mely megadja a szavazatok számát az i jelöltre vonatkozóan.

A $t_1, \ldots, t_n$ értékek kiszámítása nehéz problémának bizonyul, időbonyolultsága: $O(m^{(n-1)/2})$ (lásd a [52] dolgozatot). Ez a séma használható nagy létszámú választások esetén, ha a szervezetek a teljes Γ, Λ értéket felosztják részekre (p.l. választási kerületek).

Bibliography

[1] S. Akiyama, T. Borbély, H. Brunotte, A. Pethő, J. M. Thuswaldner, *On a generalization of the radix representation – a survey*, in "High Primes and Misdemeanours: lectures in honour of the 60th birthday of Hugh Cowie Williams", Fields Institute Commucations, **41** (2004), 19 – 27.

[2] S. Akiyama, T. Borbély, H. Brunotte, A. Pethő, J. M. Thuswaldner, *Generalized radix representations and dynamical systems I*, Acta Math. Hungar. **108** (2005), 207 – 238.

[3] S. Akiyama, H. Brunotte, A. Pethő, J. M. Thuswaldner, *Generalized radix representations and dynamical systems II*, Acta Arithmetica **121** (2006), 21 – 61.

[4] S. Akiyama, H. Brunotte, A. Pethő and J. M. Thuswaldner, *Generalized radix representations and dynamical systems III*, Osaka Journal of Mathematics **45** (2008), 347 – 374.

[5] S. Akiyama, H. Brunotte, A. Pethő, *Cubic CNS polynomials, notes on a conjecture of W.J. Gilbert*, J. Math. Anal. and Appl., **281** (2003), 402–415.

[6] S. Akiyama, H. Brunotte, A. Pethő, W. Steiner, *Remarks on a conjecture on certain integer sequences*, Periodica Math. Hungarica **52** (2006) no. 1 , 1–17.

[7] S. Akiyama, A. Pethő, *On canonical number systems*, Theoret. Comput. Sci., **270** (2002), 921–933.

[8] S. Akiyama, H. Rao, *New criteria for canonical number systems*, Acta Arith., **111** (2004), 5–25.

[9] S. Akiyama, K. Scheicher, *Symmetric shift radix systems and finite expansions*, Mathematica Pannonica, **18** (2007) no. 1, 101–124.

[10] S. Akiyama and J. M. Thuswaldner, *On the topological structure of fractal tilings generated by quadratic number systems*, Comput. Math. Appl. **49** (2005) 1439–1485.

[11] O. Baudron, P. Fouque, D. Pointcheval, G. Poupard, J. Stern, *Practical Multi-Candidate Election System*, 20th ACM Symposium on Principles of Distributed Computing ACM, (2001), 274–283.

[12] J. Benaloh, D. Tuinstra, *Receipt-free secret-ballot elections*, Proceedings of the 26th ACM Symposium on the Theory of Computing, ACM (1994), 544–553.

[13] T. Borbély, *Általánosított számrendszerek*, Master Thesis, University of Debrecen, 2003.

[14] C. Boyd, *A new multiple key cipher and an improved voting scheme*, In Advances in Cryptology - EUROCRYPT '89, LNCS Springer-Verlag, **434** (1988), 615–625.

[15] H. Brunotte, *On trinomial bases of radix representations of algebraic integers*, Acta Sci. Math. (Szeged), **67** (2001), 521 – 527.

[16] H. Brunotte, *On cubic CNS polynomials with three real roots*, Acta Sci. Math. (Szeged), **70** (2004), 495–504.

[17] H. Brunotte, A. Huszti, A. Pethő, *Bases of canonical number systems in quartic algebraic number fields*, Journal de theorie des nombres de Bordeaux, **18** (2006), 537–559.

[18] D. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital pseudonyms*, Communications of the ACM, **24** (1981), 84–90.

[19] D. Chaum, *Blind Signatures for Untraceable Payments*, In Advances in Cryptology - CRYPTO '82 Plenum Press, (1983), 199–203.

[20] D. Chaum, T. P. Pedersen, *Wallet databases with observers*, In Proc. of 12th CRYPTO Conference, LNCS Springer, **740** (1992), 3.1–3.6.

[21] R. Cramer, M. Franklin, B. Schoenmakers, M. Young, *Multi-authority secret-ballot elections with linear work*, In Advances in Cryptology - EUROCRYPT '96, LNCS Springer-Verlag, **1070** (1996), 72–83.

[22] R. Cramer, R. Gennaro, B. Schoenmakers,*A secure and optimally efficient multi-authority election scheme*, Proceedings of EUROCRYPT '97, LNCS Springer-Verlag, **1233** (1997), 103–118.

[23] Y. Desmedt and Y. Frankel, *Threshold Cryptosystems*, Advances in Cyptology-CRYPTO '89 Proceedings, Springer-Verlag (1990), 307–315.

[24] W. Diffie, M.E. Hellman, *Multiuser Cryptographic Techniques*, Proceedings of AFIPS National Computer Conference, (1976), 109–112.

[25] I. Damgard, M. Juric, *A Generalization, a Simplification and Some Applications of Pallier's Probabilistic Public-Key System*, Public Key Cryptography'01, LNCS 1992 Springer-Verlag, (2001), 119–136.

[26] T. ElGamal, *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transactions on Information Theory, **IT-31**, (1985), n. 4, 469–472.

[27] Ch. Frougny, B. Solomyak, *Finite beta-expansions*, Ergod. Th. and Dynam. Sys. **12** (1992), 713–723.

[28] A. Fujioka, T. Okamoto, K. Ohta, *A practical secret voting scheme for large scale elections*, In Advances in Cryptology - ASISACRYPT '92, LNCS Springer-Verlag, **718** (1992), 244–251.

[29] I. Gaál, *Diophantine equations and power integral bases*, Birkhäuser (Berlin), (2002).

[30] W. J. Gilbert, *Radix representations of quadratic fields*, J. Math. Anal. Appl., **83** (1981), 264–274.

[31] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, *Robust Treshold DSS Signatures*, Information and Computation, **164** (2001), Issue 1, 54–84.

[32] P. Golle, M. Jakobsson, *Reusable anonymous return channels*, Proceedings of the 2003 ACM workshop on Privacy in the electronic society ACM Press, (2003), 94–100.

[33] E. H. Grossman, *Number bases in quadratic fields*, Studia Sci. Math. Hungar., **20** (1985), 55–58.

[34] V. Grünwald, *Intorno all'aritmetica dei sistemi numerici a base negativa con particolare riguardo al sistema numerico a base negativo-decimale per lo studio delle sue analogie coll'aritmetica ordinaria (decimale)*, Giornale di matematiche di Battaglini, **23** (1885), 203–221, 367.

[35] K. Győry, *Sur les polynômes à coefficients entiers et de discriminant donné III*, Publ. Math. (Debrecen), **23** (1976), 141–165.

[36] M. Hirt, K. Sako, *Efficient receipt-free voting based on homomorphic encryption*, Proceedings of EUROCRYPT 2000, LNCS Springer-Verlag, **1807** (2000), 539–556.

[37] P. Horster, H. Petersen and M. Michels, *Meta-ElGamal signature schemes*, Proc. of the 2nd Annual ACM Conference on Computer and Communications Security ACM Press, (1994), 96–107.

[38] A. Huszti, *A secure electronic voting scheme*, Periodica Polytechnica Electrical Engineering, **51/3-4** (2007), 1–6.

[39] A. Huszti, *A Homomorphic Encryption-Based Secure Electronic Voting Scheme*, submitted for publication.

[40] A. Huszti, K. Scheicher, P. Surer, J. M. Thuswaldner, *Three-dimensional symmetric shift radix systems*, Acta Arithmetica, **129** (2007),147–166.

[41] K.R. Iversen, *A cryptographic scheme for computerized general elections*, In Advances in Cryptology - CRYPTO '91, LNCS Springer-Verlag, **576** (1992), 405–419.

[42] A. Juels, D. Catalano, M. Jakobsson, *Coercion-Resistant Electronic Elections*, Proceedings of the 2005 ACM workshop on Privacy in the electronic society, (2005), 61–70.

[43] I. Kátai, B. Kovács, *Kanonische Zahlensysteme in der Theorie der quadratischen algebraischen Zahlen*, Acta Sci. Math. (Szeged), **42** (1980), 99–107.

[44] I. Kátai, B. Kovács, *Canonical number systems in imaginary quadratic fields*, Acta Math. Acad. Sci. Hungar., **37** (1981), 159–164.

[45] I. Kátai, J. Szabó, *Canonical number systems for complex integers*, Acta Sci. Math. (Szeged), **37** (1975), 255–260.

[46] S. Körmendi, *Canonical number systems in* $\mathbb{Q}(\sqrt[3]{2})$, Acta Sci. Math. P1(Szeged), **50** (1986), 351–357.

[47] B. Kovács, *Canonical number systems in algebraic number fields*, Acta Math. Acad. Sci. Hungar., **37** (1981), 405–407.

[48] B. Kovács, *CNS rings*, Colloq. Math. Soc. János Bolyai, **34** (1981), 961–971.

[49] B. Kovács, A. Pethő, *Number systems in integral domains, especially in orders of algebraic number fields,* Acta Sci. Math. (Szeged), **55** (1991), 287–299.

[50] D. E. Knuth, *An imaginary number system*, Comm. ACM, **3** (1960), 245–247.

[51] D. E. Knuth, *The Art of Computer Programming, Vol. 2 Seminumerical Algorithms*, Addison Wesley (1998), London 3rd edition.

[52] B. Lee, K. Kim, *Receipt-free electronic voting through collaboration of voter and honest verifier*, Proceeding of JW-ISC2000 (2000), 101–108.

[53] G. Lettl, A. Pethő, *Complete solution of a family of quartic Thue equations*, Abh. Math. Sem. Univ. Hamburg **65** (1995), 365–383.

[54] E. Magkos, M. Burmester, V. Chrissikopoulos, *Receipt-freeness in large-scale elections without untappable channels*, In B.

Schmid et al., editor, First IFIP Conference on E-Commerce, E-Business, E-Government (I3E) (2001), 683–694.

[55] M. Mignotte, A. Pethő, R. Roth, *Complete solutions of quartic Thue and index form equations,* Math. Comp. **65** (1996), 341–354.

[56] T. S. Motzkin, H. Raiffa, G. L. Thompson,R. L. Thrall, *The double description method* Contributions to the theory of games, vol. 2, pp. 51-73. Annals of Mathematics Studies, no. 28. Princeton University Press, Princeton, N. J., (1953).

[57] P. Olajos, *Power integral bases in the family of simplest quartic fields*, Experiment. Math. **14** (2005), 129–132.

[58] T. Okamoto, *An electronic voting scheme*, Proceedings of IFIP '96, Advanced IT Tools Chapman & Hall, (1996), 21–30.

[59] T. Okamoto,*Receipt-Free Electronic Voting Schemes for Large Scale Elections*, Proceedings of Workshop of Security Protocols '97, LNCS Springer-Verlag, **1163** (1996), 125–132.

[60] P. Pallier, *Public-Key Cryptosystems Based on Discrete Logarithm Residues*, EUROCRYPT'99, LNCS Springer-Verlag, **1592** (1999), 223–238.

[61] C. Park, K. Itoh, K. Kurosawa, *Efficient anonymous channel and all/nothing election scheme*, In Advances in Cryptology - EUROCRYPT '93, LNCS Springer-Verlag, (1993), 248–259.

[62] W. Parry, *On the β-expansions of real numbers*, Acta Math. Acad. Sci. Hungar. **11** (1960), 401–416.

[63] T. Pedersen, *Non-interactive and information-theoretic secure verifiable secret sharing*, Proceedings of the 11th CRYPTO Conference, LNCS Springer-Verlag, **576** (1991), 129–140.

[64] S. Pemmaraju, S. Skiena, *Computational Discrete Mathematics, Combinatorics and Graph Theory with Mathematica®*, Cambridge University Press (2003).

[65] A. Pethő, *On a polynomial transformation and its application to the construction of a public key cryptosystem,* Computational Number Theory, Proc., Walter de Gruyter Publ. Comp. Eds.: A. Pethő, M. Pohst, H. G. Zimmer and H. C. Williams (1991), 31–43.

[66] A. Pethő, *Notes on CNS polynomials and integral interpolation*, More Sets, Graphs and Numbers, Eds.: E Györy, G.O.H. Katona and L. Lovász, 301–315.

[67] A. Pethő, *Connections between power integral bases and radix representations in algebraic number fields*, Proc. of the 2003 Nagoya Conf. "Yokoi-Chowla Conjecture and Related Problems", Furukawa Total Pr. Co. (2004), 115–125.

[68] D. Pointcheval, J. Stern, *Provably Secure Blind Signature Schemes*, Proceedings of ASIACRYPT '96, M. Y. Rhee and K. Kim Eds. Springer-Verlag, LNCS **1163** (1996), 252–265.

[69] I. Ray, I. Ray, N. Narasimhamurthi, *An anonymous electronic voting protocol for voting over the Internet*, Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS '01) (2001), 188–190.

[70] A. Rényi, *Representations for real numbers and their ergodic properties*, Acta Math. Acad. Sci. Hungar. **8** (1957), 477–493.

[71] R.L. Rivest, A. Shamir, L.M. Adleman, *A Method for Obtaining Digital Signatures and Public-key Cryptosystems*, Communications of the ACM, **21** (1978), n. 2, 120–126.

[72] R. Robertson, *Power bases for cyclotomic integer rings*, J. Number Theory, **69** (1998), 98–118.

[73] R. Robertson, *Power bases for 2-power cyclotomic integer rings*, J. Number Theory, **88** (2001), 196–209.

[74] K. Sako, J. Kilian, *Receipt-free mix-type voting schemes - a practical solution to the implementation of voting booth*, Proceedings of EUROCRYPT '95, LNCS Springer-Verlag, **921** (1995), 393–403.

[75] K. Scheicher, *Kanonische Ziffernsysteme und Automaten,* Grazer Math. Ber., **333** (1997), 1–17.

[76] K. Scheicher and J. M. Thuswaldner, *On the characterization of canonical number systems*, Osaka J. Math. **41**, (2004), no.2.

[77] K. Scheicher and J. M. Thuswaldner, *Digit systems in polynomial rings over finite fields*, Finite Fields Appl., **9** (2003), 322–333.

[78] I. Schur, *"Uber Potenzreihen, die im inneren des Einheitskreises beschr"ankt sind*, J. reine angew. Math. **148** (1918), 122–145.

[79] B. Schneier, *Applied Cryptography, Protocols, Algorithms and Source Code in C*, John Wiley & Sons, Inc., (1996).

[80] C. P. Schnorr, *Efficient Signature Generation for Smart Cards*, Journal of Cryptology, **4**, (1991) n. 3, 161–174.

[81] D. Shanks, *The simplest cubic fields,* Math. Comp., **28** (1974), 1137–1152.

[82] P. Surer, *Personal homepage*, Webpage `http://www.palovsky.com`

[83] P. Surer, *New characterisation results for shift radix systems*, Math. Pannon., **18** (2007), 265–297.

[84] T. Takagi, Lectures in Algebra, (1965).

[85] R. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput. **1** (1972), no. 2, 146–160.

[86] J. M. Thuswaldner, *Elementary properties of canonical number systems in quadratic fields*, in: Applications of Fibonacci Numbers, , G. E. Bergum et al. (eds.), Kluwer Academic Publishers, Dordrecht **7** (1998), 405–414.

Appendix A

List of papers of the author and citations to them

1. H. Brunotte, A. Huszti, A. Pethő, *Bases of canonical number systems in quartic algebraic number fields*, Journal de Théorie des Nombres de Bordeaux, **18** (2006), 537 – 559.

    - S. Akiyama, H. Brunotte, A. Pethő, Reducible cubic CNS polynomials, *Periodica Mathematica Hungarica*, **55** (2007), 177 – 183.

2. A. Huszti, K. Scheicher, P. Surer, J. M. Thuswaldner, *Three-dimensional symmetric shift radix systems*, Acta Arithmetica, **129** (2007), 147 – 166.

    - G. Barat, V. Berthé, P. Liardet, J. Thuswaldner, Dynamical directions in numeration, *Annales de l'institut Fourier*, **56** (2006), 1987 – 2092.

3. A. Huszti, *A Secure Electronic Voting Scheme*, Periodica Polytechnica Electrical Engineering, **51/3-4** (2007), 1 – 6.

4. J. Folláth, A. Huszti, A. Pethő, *DESignIn Asymmetric Authentication System*, Proceedings of ICAI'07 7th International Conference on Applied Informatics **1** (2007), 53 – 61.

5. A. Huszti, *A Homomorphic Encryption-Based Secure Electronic Voting Scheme*, submitted for publication.

Appendix B

List of talks of the author

1. A Secure Electronic Exam System, *International Conference on Automata, Languages and Related Topics (ALRT),* Debrecen, Hungary, 2008.
2. Secure Electronic Elections, *Cryptography and Number Theory Seminar,* Niigata University, Niigata, Japan, 2008.
3. Secure Electronic Elections, *Cryptography and Number Theory Seminar,* Nihon University, Tokyo, Japan, 2008.
4. A Secure Electronic Exam System, *Central European Conference on Cryptography,* Graz, Austria, 2008.
5. A Secure Electronic Homomorphic Voting Scheme, *International Conference on Applied Informatics,* Eger, Hungary, 2007.
6. A Secure Electronic Voting Scheme Based on Blind Signatures, *Conference of PhD Students in Computer Science,* Szeged, Hungary, 2006.
7. A Secure Electronic Voting Scheme Based on Blind Signatures, *NyírCrypt Central European Conference on Cryptography,* Nyíregyháza, Hungary, 2006.
8. Canonical Number Systems in Quartic Number Fields, *Number Theory Seminar,* Montanuniversitet, Leoben, Austria, 2005.

Appendix C

Acknowledgements

I would like to thank Attila Pethő for being an excellent supervisor. I am grateful about his numerous valuable hints and suggestions.

In addition I want to thank László Csirmaz for the helpful answers to specific questions I asked from him. The discussions with him always inspire me.

Thanks Horst Brunotte, Klaus Scheicher, Paul Surer and Jörg M. Thuswaldner for the collaborative work on research papers that are included in this dissertation.

I would like to thank Number Theory Research Group of Hungarian Academy of Sciences and University of Debrecen for the financial support.

Last, but not least, I thank my father, my mother and my husband for their love and patience throughout working on my thesis.

Appendix D

Köszönetnyilvánítások

Köszönöm Pethő Attilának, hogy kiváló témavezetőm volt. Hálás vagyok a számos, iránymutató ötletéért és javaslatáért.

Szeretném megköszönni Csirmaz Lászlónak a speciális kérdéseimre adott sok segítséget nyújtó válaszait. A vele való beszélgetések mindig lelkesítenek.

Köszöm Horst Brunotte, Klaus Scheicher, Paul Surer és Jörg M. Thuswaldner társszerzőim kooperatív munkáját, hogy segítettek a disszertációban szereplő cikkek elkészítésében.

Köszöm a Magyar Tudományos Akadémia Debreceni Számelméleti Kutatócsoportjának az anyagi támogatást.

Végül, de nem utolsó sorban, köszönöm édesapám, édesanyám és férjem végtelen türelmét és szeretetét, támogatásukkal lehetővé tették e disszertáció elkészülését.