Université de Limoges

Thèse de Doctorat-École Doctorale 610
Spécialité: Mathématiques

On Certain Types of Code-Based Signatures

Soutenue le 30/11/2021 par
Dang Truong Mac

JURY

Philippe Gaborit	Professfeur, Université de Limoges	Directeur
Duong Hieu Phan	Professeur, Techcom Paris, Institut Polyechnique de Paris	Co-directeur
Olivier Blazy	Professeur, École Polytechnique	Examinateur
Carlos Aguilar Melchor	Professeur, Isae Supareo, Université de Toulouse	Examinateur
Alain Couvreur	Directeur de Recherche, INRIA Saclay	Rapporteur
Pascal Lafourcade	Maître de Conférence (HDR), Université de Clermont Auvergne	Rapporteur

To My Parents

Acknowledgements

I would like to thank Fate for having united me and my supervisors, for giving me a chance to study Cryptography and, particularly, Code-Based Cryptography.

I would like to thank my thesis supervisors Prof. Philippe Gaborit and Prof. Duong Hieu Phan. I would like to thank Dr. Olivier Blazy also. During my internship and especially my PhD period, I have learned a lot from them.

Next, I would like to express my thanks to the staffs of XLIM especially the secretaries of the MATHIS department. I received much help and instructions while registering to XLIM and also in other procedures.

To my friend, I would like to thank Quoc Huy Vu for his warming helps, entertainments. We journeyed together and discussed lots of problems with each other.

Finally, I wish to express my innermost gratitude to my family.

Abstract

Digital signatures were first introduced in the work of Diffie and Hellman, dated back in 1976. It is a scientific art replacing the traditional way of written signatures. Each signer has a "personal knowledge," or a signing key, to produce signatures. And as the same as handwritten signatures, anyone seeing this signature would be convinced that it belong to a certain person (and no one else). In order to produce such a signature, the signing key is indispensable, and the secret of this entity is usually protected by the hardness assumption of some computational problems. In the earliest stage, these are number theoretic problems such as factoring large integer numbers or computing the discrete logarithm of an element with respect to some prime modulus. However, with the rapid development of technology, these problems will be solved efficiently when the era of quantum computer arrives. Then comes the next stage in the progressing course of digital signatures when most of the attention is given to the decoding problem (and many of its variants), of which the hardness resists even the quantum computer. This problem, however, takes part in two important branches of cryptography, namely, lattice-based cryptography and code-based cryptography due to the main object it is related to.

This thesis mainly concerns with signatures in the latter branch, *i.e.,* the code-based cryptography. It proposes two main contributions.

The first of which is a signature scheme in the Hamming metric context. The scheme is achieved as an application of a chameleon hash function, which is constructed entirely from classical code-based hardness assumptions. The most notable feature of this scheme is that it is proved to be secure in the standard model. While security of code-based schemes in the random oracle model is still unclear, such property is highly desirable.

The second contribution is a group signature scheme in the rank metric context. In general, the construction of the scheme follow the frame devised for the Hamming metric. At the core, this frame uses two permutations which are designed from a random vector. Though quite efficient for the binary case, that is, the base field is $\mathbb{F}_2$, this method shows its disadvantages when the base field is changed. A natural question arises out of this situation: How can we construct schemes

in another fields? We answer this question by proposing a different method of permuting. Our method has the advantage that it can be applied regardless the metric being in consideration.

Résumé

Les signatures numériques ont été introduites pour la premiere fois dans les travaux de Diffie et Hellman en 1976. C'est un art scientifique remplaçant la méthode traditionnelle des signatures écrites. Chaque signataire possède un "secret personnel ", aussi appelé clé de signature, pour produire des signatures. Tout comme les signatures manuscrites, chaque signature numérique est unique et peut être rattachée à la personne qui l'a signée aux yeux d'un observateur. Afin de produire une telle signature, la clé de signature est indispensable, et le secret de cette clé est généralement protégé par une hypothèse difficile de certains problèmes calculatoires. Parmi les problèmes possibles, on peut citer par exemple en théorie des nombres, la factorisation de grands entiers ou le calcul d'un logarithme discret dans un module premier. Cependant, ces problèmes seront résolus efficacement lorsque l'ère de l'ordinateur quantique arrivera. On peut alors se tourner vers d'autres types de problèmes, qu'on pourrait qualifier comme étant des problèmes de décodage (et de leurs variantes), qui résistent à l'ordinateur quantique. Ces problèmes font partie de deux branches importantes de la cryptographie, à savoir la cryptographie basée sur les réseaux et la cryptographie basée sur les codes correcteurs d'erreur.

Cette these concerne principalement les signatures basées sur des problèmes dans cette dernière branche, à savoir la cryptographie basée sur les codes correcteurs d'erreur. Elle propose deux contributions dans ce domaine.

La première est un schema de signature dans la métrique de Hamming. Ce schema résulte d'une fonction de hachage caméléon qui est construit à partir des problèmes difficiles de code. La caractéristique la plus notable de ce schema est qu'il s'avère sûr dans le modèle standard. Bien que la sécurité des schémas basés sur les codes dans le modèle d'oracle aléatoire ne soit pas toujours claire, une telle propriété est hautement souhaitable.

La seconde contribution est un schéma de signature de groupe basé sur la métrique rang. En général, la construction d'un schéma de ce type suit plutôt le cadre conçu pour la métrique de Hamming. Essentiellement, ce cadre utilise deux permutations qui sont conçues a partir d'un vecteur aléatoire. Bien qu'assez efficace pour le cas binaire, c'est à dire dans le corps $\mathbb{F}_2$, les inconvénients de cette

méthode se révèlent lorsque le corps de base est modifié. Une question naturelle surgit dans cette situation : Comment pouvons-nous construire des schémas dans d'autres corps ? Nous répondons à cette question en proposant une méthode différente de permutation. Notre méthode a l'avantage de pouvoir être appliquée quelle que soit la métrique considérée.

Contents

List of Symbols

$\mathbb{N}$	the set of positive integers
$\mathbb{R}$	the set of real numbers
$\mathbb{F}_q$	the finite field of q elements
$\lvert X \rvert$	the cardinality of the set X
$\lvert x \rvert$	the length of string x
$\varnothing$	the empty set
$\circ$	the composition operation
$\mathbf{v}^T$	the transposed vector of vector $\mathbf{v}$
$\mathbf{A}^T$	the transposed matrix of matrix $\mathbf{A}$
$\begin{bmatrix} n \\ k \end{bmatrix}_q$	GAUSSian coefficient
$\mathcal{S}_w^n$	the sphere centered at $\mathbf{0}$ of radius w in $\mathbb{F}_q^n$
$\mathcal{S}_w^{n,m}$	the sphere centered at $\mathbf{0}$ of radius w in $\mathbb{F}_{q^m}^n$
$\mathrm{GL}(n, q)$	the set of invertible matrices in $\mathbb{F}_q^{n \times n}$

Chapter 1

Introduction

1.1 Motivations

In 1994, Peter Shor [Sho94] introduced the first quantum algorithm to factorize integer numbers. This algorithm runs in polynomial time which means that most of the cryptosystems such as RSA [RSA78], ElGamal [ElG84] would become insecure when the quantum computer era comes. Since then, numerous post-quantum cryptosystems have been devised.

The activity was really blooming out after the breakthrough work of Ajtai [Ajt96]. The hardness of lattice problems such that the short integer solution problem (and its variants) and sepecially the learning with error problem are studied thoroughly with the typical works of Regev [Reg05], Micciancio and Regev [MR04], and many others. Together with these studies are schemes and constructions which plays an essential role for new coming schemes. Particularly in the signature aspect are the hash-and-sign scheme of Gentry *et al.* [GPV08] (GPV for short) and the signing without trapdoor scheme of Lyubashevsky [Lyu12]. Lattice-based cryptography has gained much of favor while it seems not the case for cryptography based on coding theory.

The history of code-based cryptography can be said to begin in 1978 with the invention of Robert McEliece–the famous McEliece cryptosystem [McE78]. (This scheme had been invented even before Schor's algorithm!) The scheme uses a structured Goppa code which is scrambled to encrypt message. The decoding algorithm of this code is the key for the decryption algorithm. The difficulties of attacking this cryptosystem lie in the hardness of solving the underlying decoding problem and in the indistinghuishability of Goppa codes and random codes. Despite almost 50 years of attacking effort, this scheme is still surviving. The only criticism this scheme receives is that its key size is comparatively large (compared to number-theoretic schemes). However, in the near future, with the development

of technology, this problem would be easily resolved.

There have many attempts to improve McEliece scheme on the key size aspect. In 2005, Gaborit [Gab05] proposed to use quasi-cyclic codes instead of Goppa codes. The structure of quasi-cyclic codes serves as great advantage in decreasing the key size. The public key now does not contain a whole matrix but just a few row vectors. This method was also used in [MTSB12] with additional properties on the parity-check matrix of the codes, *i.e.,* the codes being used are (quasi) moderate density parity-check codes (MDPC). Also in rank metric, several constructions are built in this spirit. The first of this line is the construction in [GMRZ13] which uses low rank parity-check codes (LRPC) and can be viewed as a rank equivalent version of the MDPC construction. And most recently is the rank quasi-cyclic cryptosystem [ABD+16] in which, first, quasi-cyclic codes, and then, ideal codes are used. It seems to be an appropriate place to mention here that its twin, the Hamming quasi-cyclic cryptosystem, also constructed on the same principle, has found its way to appear in the 3rd round of the NIST's call for post-quantum cryptography.

In the aspect of signature, the first notable one is the Stern's identification protocol [Ste94]. Briefly speaking, this scheme allows one party to convince other party of the fact that the former possesses some secret information without giving this information to the latter. Though this is not a signature scheme, however, when being combined with the Fiat-Shamir transform [FS87], one can produce signatures. The signature is a proof of knowledge on the secret information. Security of signatures of this type is tightly reduced to the well-known difficult problems in coding theory. Stern's idea has its impact even in lattice-based cryptography, which is visible in the work of Kawachi *et al.* [KTX08] and Ling *et al.* [LNSW13]. Recent development of Stern's protocol in the code-based field is a concatenated version and could be found in the work of [ABCG16a]. Stern's identification scheme has shown itself to be extremely versatile.

The second construction which also has a great impact is the signature scheme designed by Kabatiansky, Krouk, and Smeets [KKS97]. The interesting feature of this scheme is that it allows a signer to sign a message without the use of any decoding algorithm. Two pieces of information are needed for the designing of such scheme, namely, lower bounds of the probability that a random linear code has the minimum distance at least d and the probability that a random linear code lies between two spheres of prescribed radii. Though the initial parameters and those of other variants such that [KKS05] and [BMS11] are all broken, the hardness assumption is still intact. This means that there are still mysterious aspects about this scheme waiting for exploring. All existing schemes are in Hamming metric, thus, one could wonder what does a scheme in rank metric look like?

Next comes the scheme designed by Courtois, Finiasz, and Sendrier (CFS)

[CFS01]. In the essence, the scheme is based on the McEliece encryption scheme (or more precisely, on the Niederreiter encryption scheme)[1]. The difference is that CFS scheme allows less errors than the classical McEliece. With this setting, finding a preimage, *i.e.,* a signature, for a message would be feasible. This scheme has been generalized as in [Fin11] and [BCMN11] and also used as a building-block in another constructions such as [ABCG16a] or [BGSS17].

Recently, progress has been made with two proposals: Durandal [ABG+19], a digital signature scheme in rank metric, and WAVE [DST19]. The former is an adaptation of Luybaschevsky's scheme mentioned above in a suitable way and the latter is constructed from a code-based trapdoor pre-image function, thus is in the line of GPV. There is no doubt that these scheme would still have many hidden properties to explore and, in particular, how they can be applied for the designing of other constructions.

All in all, code-based cryptography has shown that it is a fruitful field of research; code-based schemes have parameters which are comparable with those of other schemes in other cryptography's branches, especially lattice-based cryptography, and above all, there are still many deep secrets in itself.

1.2 Contributions and Organization

The main part of this thesis resides in Chapter 3 and 4, which present two of our contributions. The third contribution is in Chapter 5. The rest of this thesis is organized in the following way.

- Chapter 2 provides basic notions on coding theory and modern cryptography. The fundamental code-based hardness assumptions are also provided. This chapter is concluded with some cryptosystems constructed based on these assumptions. These protocols, on one hand, can be regarded as examples for the notions and assumptions just provided and, on the other hand, will serve as preliminaries to our works.

- Chapter 3 presents our first contribution to post-quantum cryptography. At the core, it is a chameleon hash function which is constructed entirely from code-based hardness assumptions. As an application, we construct a digital signature scheme in the hash-and-sign paradigm, where the ordinary hash function is replaced by the chameleon hash function.

- Chapter 4 describes our second contribution, a static group signature scheme in the rank metric context. The scheme makes use of the RQC cryptosystem

[1]The Niederreiter cryptosystem is usually regarded as dual to the McEliece's whereas parity-check matrix is used instead of generator matrix.

and a signature scheme in the Stern's frame. These two pieces are "glued" together by a zero-knowledge protocol. The task of designing such a protocol is handled by two permutations which are derived from the multiplication operation of polynomials.

- Chapter 5 is a minor contribution compared to the two in the above chapters; its is a reparation of the blind signature scheme which is designed in [BGSS17]. First, the flaw will be precised. Briefly speaking, it is a lack in the construction and hence, the collision problem as well as the use of a successful adversary in the unforgeability proof cannot be efficiently handled. Then, comes the reparation scheme, which differs from the previous by adding a proof of knowledge of the used randomness.

- Chapter 6 will conclude this thesis. It recaptures the whole thesis and draws a sketch of future works.

Chapter 2

Prerequisites

2.1 Hamming Metric Codes

2.1.1 Linear Codes

Let q be a power of a prime number and $\mathbb{F}_q$ the finite field of q elements. A linear code of length n over $\mathbb{F}_q$ is a subset $\mathcal{C}$ of $\mathbb{F}_q^n$ such that $\mathbf{0} \in \mathcal{C}$, and for all $\mathbf{x}, \mathbf{y} \in \mathcal{C}$, the sum $\mathbf{x} + \mathbf{y}$ is also in $\mathcal{C}$. With these properties, such a set $\mathcal{C}$ is indeed a vector subspace of $\mathbb{F}_q^n$. Let k be the dimension of $\mathcal{C}$ and $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ a set of linearly independent elements of $\mathcal{C}$ over $\mathbb{F}_q$, then each element $\mathbf{c}$ of $\mathcal{C}$ is uniquely represented as

$$\mathbf{c} = x_1\mathbf{c}_1 + \cdots + x_k\mathbf{c}_k, \tag{2.1}$$

where $x_1, \ldots, x_k \in \mathbb{F}_q$. Now, let $\mathbf{G}$ be the matrix whose rows are the $\mathbf{c}_i$'s, then Equation 2.1 can be rewritten as $\mathbf{c} = (x_1, \ldots, x_k) \cdot \mathbf{G}$. The matrix $\mathbf{G}$ is called a *generator matrix* of the code $\mathcal{C}$, and from now on, the elements of $\mathcal{C}$ are called the *codewords* of $\mathcal{C}$. (Other elements of $\mathbb{F}_q^n$, which are not in $\mathcal{C}$, are usually called *words*.) Thus, from generator matrix, a linear code can be defined as follows.

Definition 2.1. *Let* $\mathbf{G}$ *be a* $k \times n$ *matrix of rank* k *with entries in* $\mathbb{F}_q$*. Define*

$$\mathcal{C} = \{\mathbf{x} \cdot \mathbf{G} \mid \mathbf{x} \in \mathbb{F}_q^k\}.$$

Then $\mathcal{C}$ *is called a* linear (n, k) code *over* $\mathbb{F}_q$*.*

Example 1. *Consider the case* $q = 2$*, and let*

$$\mathbf{G} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

The code $\mathcal{C}$ *generated by* $\mathbf{G}$ *is of length* $n = 4$ *and consists of* 4 *codewords. That is,*

$$\mathcal{C} = \big\{(0,0,0,0); (0,1,0,1); (1,0,1,0); (1,1,1,1)\big\}.$$

For a given matrix $\mathbf{G}$, let $\mathbf{H}$ be the $(n-k) \times n$ matrix such that $\mathbf{H} \cdot \mathbf{G}^T = 0$. From this equation, one can easily see that for any codeword $\mathbf{c}$ of the code defined by $\mathbf{G}$,

$$\mathbf{H} \cdot \mathbf{c}^T = \mathbf{0}^T$$

holds true. Therefore, a linear code can also be defined as follows.

Definition 2.2. *Let* $\mathbf{H}$ *be an* $(n-k) \times n$ *matrix whose entries are elements of* $\mathbb{F}_q$. *The following set*

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n \mid \mathbf{H} \cdot \mathbf{c}^T = \mathbf{0}^T\}$$

is called a linear (n, k) code *over* $\mathbb{F}_q$. *The parameter* n *is usually referred to as the* length, *the parameter* k *the* dimension *of* $\mathcal{C}$. *The matrix* $\mathbf{H}$ *is called a* parity-check matrix *of* $\mathcal{C}$.

Example 2. *One can easily verify that the matrix* $\mathbf{G}$ *of Example 1 satisfies*

$$\mathbf{G} \cdot \mathbf{G}^T = 0.$$

Therefore, the matrix $\mathbf{G}$ *can also be viewed as a parity-check matrix of* $\mathcal{C}$. *Such a code is called* self-dual *code.*

The origin of linear codes springs from the purpose of communication through a "noisy" channel. In reality, a message is a string of 0's and 1's, which are the elements of the field of two elements $\mathbb{F}_2$. More generally, a message can be a string of symbols that are elements of a finite field. Now, such a message is transmitted. Due to the fact that the channel is not ideal, some symbols of the message may be altered and thus, the received message is not the same as the transmitted one. This leads to the problem: how to recover the original message from the received one. That is where linear codes play their role by offering a solution. The idea is that the message is stretched in a particular way before transmitted.

Let $(a_1, \ldots, a_k)$ be the message, where a_i's are the elements of some finite field $\mathbb{F}_q$. This message is stretched or *coded* into a codeword $\mathbf{c} = (c_1, \ldots, c_n)$ of some code $\mathcal{C}$ over $\mathbb{F}_q$. The essential constraint here is that $n > k$. The codeword $\mathbf{c}$ suffers from the alteration caused by a noisy channel and is changed into $\mathbf{c} + \mathbf{e}$. This is the received message, and after the recovering process, which is *decoding*, one gets the message $(a_1, \ldots, a_k)$. This whole process is best illustrated by the following figure.

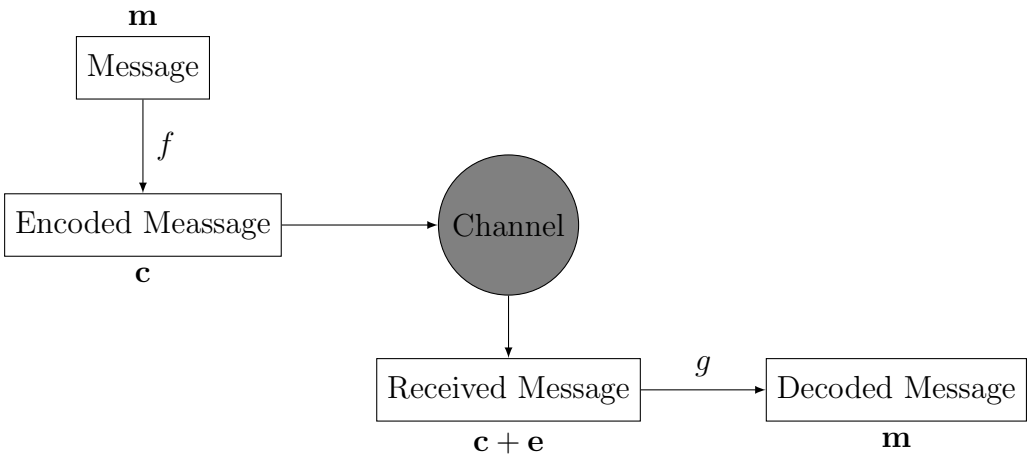

Figure 2.1: A communication system.

Besides its length and dimension, one is usually concerned with the minimal distance of a code. For two codewords $\mathbf{x}, \mathbf{y}$ of a code $\mathcal{C}$, the HAMMING distance between $\mathbf{x}$ and $\mathbf{y}$ is defined to be equal to the number of nonzero coordinates of the codeword $\mathbf{x} - \mathbf{y}$. Letting $\mathbf{y} = \mathbf{0}$, one gets the distance between $\mathbf{x}$ and $\mathbf{0}$, or the HAMMING *weight* of $\mathbf{x}$. These notions are rephrased in the following definition.

Definition 2.3. *Let $\mathbf{x}, \mathbf{y}$ be two codewords of a linear code $\mathcal{C}$ over $\mathbb{F}_q$. Then*

(*i*) *the distance between $\mathbf{x}$ and $\mathbf{y}$, denoted by $d(\mathbf{x}, \mathbf{y})$, is the number of nonzero coordinates of $\mathbf{x} - \mathbf{y}$;*

(*ii*) *the weight of $\mathbf{x}$, denoted by $w(\mathbf{x})$ or sometimes by $\|\mathbf{x}\|$, is the number of nonzero coordinates of $\mathbf{x}$.*

Since a linear code $\mathcal{C}$ is a subset of $\mathbb{F}_q^n$, so the above definition is, in fact, applied for $\mathbb{F}_q^n$. The HAMMING distance has the following basic properties.

Proposition 2.1. *Let $\mathbf{x}, \mathbf{y}, \mathbf{z}$ be vectors of $\mathbb{F}_q^n$. Then*

(*i*) $d(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$;

(*ii*) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$;

(*iii*) $d(\mathbf{x}, \mathbf{y}) + d(\mathbf{y}, \mathbf{x}) \geq d(\mathbf{x}, \mathbf{z})$.

Proof. The first two statements clearly come from the definition. For (*iii*), let k be a position where $x_k \neq y_k$. The statement follows from the fact that either $x_k \neq z_k$ or $y_k \neq z_k$. □

The above proposition shows that HAMMING distance is indeed a metric on $\mathbb{F}_q^n$. The minimum distance of a code $\mathcal{C}$ is the least positive number $d_\mathcal{C}$ such that $d(\mathbf{x}, \mathbf{y}) \geq d_\mathcal{C}$ for all codewords $\mathbf{x}, \mathbf{y}$ of $\mathcal{C}$. Since $\mathbf{0}$ is a codeword of $\mathcal{C}$, so the minimum distance of $\mathcal{C}$ is indeed the minimum weight of nonzero codewords of $\mathcal{C}$, *i.e.,*

$$d_\mathcal{C} = \min_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} \{w(\mathbf{c})\}.$$

The minimum distance of a code characterizes its capacity of decoding. Assume that $\mathbf{c}$ is a transmitted codeword through a "noisy" channel and one receives a word $\mathbf{y}$. Then $\mathbf{y}$ is of the form $\mathbf{c} + \mathbf{e}$ for some small weight word $\mathbf{e}$ of $\mathbb{F}_q^n$. The rule to decode $\mathbf{y}$ is to find *the* codeword $\mathbf{c}$ that minimizes $w(\mathbf{y} - \mathbf{c})$, or in other words, to find the *nearest* codeword of $\mathcal{C}$ to $\mathbf{y}$. This rule is called *nearest neighbour decoding* [LN94].

Definition 2.4. *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a linear code and t a positive integer. Then $\mathcal{C}$ is called* t-error-correcting *if for any $\mathbf{y} \in \mathbb{F}_q^n$, there is at most one codeword $\mathbf{c} \in \mathcal{C}$ such that $d(\mathbf{c}, \mathbf{y}) \leq t$.*

From the above definition, one sees at once that a linear code $\mathcal{C}$ with minimum distance $d_\mathcal{C} \geq 2t + 1$ is t-error-correcting. Indeed, for a vector $\mathbf{x} \in \mathbb{F}_q^n$, let

$$B_t(\mathbf{x}) = \{\mathbf{v} \in \mathbb{F}_q^n \mid d(\mathbf{x}, \mathbf{v}) \leq t\}$$

be the ball of radius t centered at $\mathbf{x}$. Because of the property that $d_\mathcal{C} \geq 2t+1$, such a ball contains at most one codeword of $\mathcal{C}$; otherwise, let $\mathbf{c}_1, \mathbf{c}_2$ be two codewords which are in the same ball $B_t(\mathbf{x})$, then

$$2t + 1 \leq d(\mathbf{c}_1, \mathbf{c}_2) \leq d(\mathbf{c}_1, \mathbf{x}) + d(\mathbf{c}_2, \mathbf{x}) \leq 2t,$$

which is a contradiction. The minimum distance of a code can be determined by simple observation as in the following lemma.

Lemma 1. *Let $\mathcal{C}$ be an (n, k)-linear code and $\mathbf{H}$ a parity-check matrix of $\mathcal{C}$. Then, $d_\mathcal{C} \geq d$ if and only if any $d - 1$ columns of $\mathbf{H}$ are linearly independent.*

Proof. Let $\mathbf{h}_1, \ldots, \mathbf{h}_n$ be the columns of $\mathbf{H}$, and consider the following equation

$$x_1\mathbf{h}_1 + \cdots + x_n\mathbf{h}_n = \mathbf{0}. \tag{2.2}$$

Assume that $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{F}_q^n$ be a solution of Equation 2.2, and thus a codeword of $\mathcal{C}$. The statement of the lemma comes from the fact that $c_{i_1}, \ldots, c_{i_j}$ are all nonzero coordinates of $\mathbf{c}$ for $i_1, \ldots, i_j \in \{1, \ldots, n\}$ means that $\mathbf{h}_{i_1}, \ldots, \mathbf{h}_{i_j}$ are linearly dependent. □

The inequality in Lemma 1 becomes an equality when the smallest sets (there may have more than one such sets) of linearly dependent columns of $\mathbf{H}$ have cardinality $d+1$. Let S be such a set and $\mathbf{h}_i$ an arbitrary element of S. By the minimality, the set $S \setminus \{\mathbf{h}_i\}$ consists of linearly independent columns of the matrix $\mathbf{H}$, and since the code $\mathcal{C}$ is of dimension k, so $\operatorname{rank} \mathbf{H} = n-k$. From the last equality, one concludes that $|S \setminus \{\mathbf{h}_i\}| \leq n-k$ or $|S| \leq n-k+1$. Consequently, one gets a simple upper-bound for the minimum distance of $\mathcal{C}$, *i.e.,* $d_{\mathcal{C}} \leq n-k+1$. This bound is commonly known as the SINGLETON bound [Sin64].

Theorem 2.1 (SINGLETON Bound)**.** *Let $\mathcal{C}$ be an (n,k)-linear code over $\mathbb{F}_q$. Then the minimum distance of $\mathcal{C}$ satisfies $d_{\mathcal{C}} \leq n-k+1$.*

Example 3. (*i*) *The code $\mathcal{C}$ in Example 2.1 has minimum distance $d_{\mathcal{C}} = 2$.*

(*ii*) *Consider the following matrix*

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Let $\mathcal{C}_1$ be the code whose parity-check matrix is $\mathbf{H}$. Then $\mathcal{C}_1$ is a $(6,4)$ linear binary code. It is easy to check that the sum of any two columns is nonzero. The last row is the all-one vector, hence the sum of any three columns is also nonzero. One sees that $\mathbf{h}_2+\mathbf{h}_3+\mathbf{h}_4+\mathbf{h}_5 = \mathbf{0}$. Therefore, the minimum distance is $d_{\mathcal{C}_1} = 4$.

In the following paragraph, a simple idea to decode linear codes is described. As above, let $\mathcal{C}$ be an (n,k)-linear code over $\mathbb{F}_q$. Assume that the cosets of $\mathcal{C}$ are $\mathbf{a}_0+\mathcal{C}, \ldots, \mathbf{a}_m+\mathcal{C}$, where $\mathbf{a}_0, \ldots, \mathbf{a}_m$ are elements of $\mathbb{F}_q^n$ and $m = q^{n-k}-1$. Since

$$\mathbb{F}_q^n = \big(\mathbf{a}_0+\mathcal{C}\big) \cup \cdots \cup \big(\mathbf{a}_m+\mathcal{C}\big),$$

so each word $\mathbf{w}$ of $\mathbb{F}_q^n$ belongs to a unique coset of $\mathcal{C}$. Therefore, a received word $\mathbf{y}$ must also lie in a unique coset. One sees that if $\mathbf{c}$ is the transmitted codeword, then the error $\mathbf{e} = \mathbf{y}-\mathbf{c}$ and the received word $\mathbf{y}$ must be in the same coset. Among the possible values for $\mathbf{e}$, the most likely one is the one with minimum weight. (Channels are designed so that the probability of noise is as small as possible.) Hence, the word $\mathbf{y}$ is decoded as $\mathbf{y}-\mathbf{e}$. Such a word $\mathbf{e}$ with minimum weight is usually referred to as a *coset leader* of the coset of $\mathbf{y}$. And as same as describing a code, there are two ways to determine the coset of a word $\mathbf{y}$. The obvious way is simply adding the codewords of $\mathcal{C}$ to $\mathbf{y}$, *i.e.,* the coset of $\mathbf{y}$ is the

set $C_{\mathbf{y}} = \mathbf{y} + \mathcal{C}$. The second way is via a parity-check matrix $\mathbf{H}$ of the code $\mathcal{C}$, that is

$$C_{\mathbf{y}} = \left\{\mathbf{x} \in \mathbb{F}_q^n \mid \mathbf{H}\mathbf{x}^T = \mathbf{H}\mathbf{y}^T\right\}.$$

This way of determining coset leads to the notion of syndrome of a word. It is defined as following.

Definition 2.5. *Let* $\mathbf{H}$ *be a parity-check matrix of an* (n, k)*-linear code* $\mathcal{C}$ *and* $\mathbf{y}$ *a word in* $\mathbb{F}_q^n$*. The vector* $\mathbf{H}\mathbf{y}^T$ *is called the* syndrome *of* $\mathbf{y}$*.*

By this definition, the coset of $\mathbf{y}$ can be described as the set of vectors whose syndromes are the same as $\mathbf{y}$'s.

2.1.2 Cyclic and Quasi-cyclic Codes

Two classes of linear codes, which are used intensely in cryptography, are cyclic codes and quasi-cyclic codes. (The latter can be regarded as a generalisation of the former.) Their algebraic structure is quite simple, and closely connected with polynomials over a finite field.

We start with cyclic codes. Suppose that we have a word $\mathbf{a}^{(0)} = (a_0, \ldots, a_{n-1}) \in \mathbb{F}_q^n$. By repeatedly doing the right-shift, one entry at a time, we get other $n-1$ words, namely, $\mathbf{a}^{(i)} = (a_i, a_{i+1}, \ldots, a_{n-1-i})$ for $i = n-1, n-2, \ldots, 0$.[1] Here, the total number of different words may be less than n and the count is taken with multiplicity. These words, by the nature of the operation, can be called the *right-shift* words of $\mathbf{a}^{(0)}$. If a linear code $\mathcal{C}$ over $\mathbb{F}_q$ has the property that $\mathbf{a} = \mathbf{a}^{(0)} \in \mathcal{C}$ implies that $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(n-1)}$ are also the codewords of $\mathcal{C}$, then $\mathcal{C}$ is called a cyclic code. This can be summarized in the following definition.

Definition 2.6. *Let* $\mathcal{C}$ *be an* (n, k)*-linear code over* $\mathbb{F}_q$*. If for any codeword* $\mathbf{a}$ *of* $\mathcal{C}$*, all of its right-shift words are also in* $\mathcal{C}$*, then* $\mathcal{C}$ *is called a cyclic code.*

Let $(x^n - 1)$ denote the ideal of $\mathbb{F}_q[x]$ generated by $x^n - 1$. Elements of $\mathbb{F}_q^n$ are mapped to the polynomial ring $\mathbb{F}_q[x]/(x^n - 1)$ by the following map

$$\begin{aligned} \phi\colon \mathbb{F}_q^n &\longrightarrow \mathbb{F}_q[x]/(x^n - 1) \\ (a_0, \ldots, a_{n-1}) &\longmapsto a_0 + \cdots + a_{n-1}x^{n-1}. \end{aligned}$$

This map is an isomorphism and hence each word of $\mathbb{F}_q^n$ can be thought of as a polynomial of degree less than n, and vice-versa. Now, let $\mathbf{a}^{(0)} = (a_0, \ldots, a_{n-1})$ be a word of $\mathbb{F}_q^n$ and $a(x) = \phi(\mathbf{a}^{(0)})$. It is not hard to see that the i-th right-shift word $\mathbf{a}_i$ of $\mathbf{a}_0$ satisfies that

$$\phi(\mathbf{a}^{(i)}) = x^{n-i}a(x).$$

[1] The indices are taken modulo n.

From this observation, a class of (n, k)-cyclic code can be defined as in the following proposition.

Proposition 2.2. *Let $g(x) = g_0 + \cdots + g_{n-k}x^{n-k}$ be a divisor of $x^n - 1$ and*

$$\mathbf{G} = \begin{pmatrix} g(x) \\ xg(x) \\ \vdots \\ x^{k-1}g(x) \end{pmatrix} = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 & 0\cdots & 0 \\ 0 & g_0 & g_1 & \cdots & g_{n-k} & 0\cdots & 0 \\ & & \ddots & & \ddots & & \\ 0 & 0 & \cdots & g_0 & g_1 & \cdots & g_{n-k} \end{pmatrix}.$$

Then the code $\mathcal{C}$ generated by $\mathbf{G}$ is an (n, k)-cyclic code.

Such a code $\mathcal{C}$ as in the above proposition is said to be generated by $g(x)$. The next proposition shows that this class, indeed, contains all (n, k)-cyclic codes of $\mathbb{F}_q^n$.

Proposition 2.3. *Let $\mathcal{C}$ be an (n, k)-cyclic code over $\mathbb{F}_q$. Then there exists a polynomial $g(x)$ of degree $n - k$ such that*

$$\phi(\mathcal{C}) = \big(g(x)\big).$$

Proof. We think of $\mathcal{C}$ as a subset of $\mathbb{F}_q[x]/(x^n - 1)$. Thus, we need to prove that there exists a polynomial $g(x)$ of degree $n - k$ such that $\mathcal{C} = \big(g(x)\big)$.

As above, we saw that if $a(x) \in \mathcal{C}$, then $x^i a(x) \in \mathcal{C}$ for $i = 1, \ldots, n - 1$. Since $\mathcal{C}$ is a linear code, this means that $b(x)a(x) \in \mathcal{C}$ for all $b(x) \in \mathbb{F}_q[x]/(x^n - 1)$.

Now, let $g(x) \in \mathcal{C}$ be the monic polynomial, *i.e.,* with leading coefficient equal to 1, whose degree is minimal. Such polynomial exists since $|\mathcal{C}|$ is finite. Obviously, we have $\big(g(x)\big) \subseteq \mathcal{C}$. If there were a polynomial $g_1(x) \in \mathcal{C} \setminus \big(g(x)\big)$, then, by linearity, $r(x) = \gcd(g, g_1)$ would be in $\mathcal{C}$. This is a contradiction, since $\deg r(x) < \deg g(x)$. Therefore, $\mathcal{C} = \big(g(x)\big)$.

Finally, let $d = \deg g(x)$. Since $\mathcal{C} = \big(g(x)\big)$ is an ideal of $\mathbb{F}_q[x]/(x^n - 1)$ so $g(x)$ must be a divisor of $x^n - 1$. On the other hand, since $\dim \mathcal{C} = k$ and $\mathcal{C} = \big(g(x)\big)$, so $g(x), \ldots, x^{k-1}g(x)$ are linearly independent over $\mathbb{F}_q$. Let $h_0, \ldots, h_{k-1}$ be elements of $\mathbb{F}_q$, not all of which are equal to 0, such that

$$h_0 g(x) + \cdots + h_{k-1}x^{k-1}g(x) + x^k g(x) = 0. \tag{2.3}$$

Define $h(x) = h_0 + \cdots + h_{k-1}x^{k-1} + x^k$, then Equation 2.3 means that

$$h(x)g(x) \equiv 0 \mod x^n - 1.$$

Compare the degrees of both sides, we have $k \geq n - d$. Let $p(x) = \frac{x^n - 1}{g(x)}$, then $\deg p(x) = n - d$, and thus $g(x), \ldots, x^{n-d}g(x)$ are linearly independent. From this, we deduce that $n - d \geq k$, and therefore, $k = n - d$ or $\deg g(x) = n - k$. □

A linear code can be defined through parity check matrix, thus one can think of a similar way to define a cyclic code via polynomial, which generates parity-check matrix. Let $\mathcal{C}$ be the cyclic code generated by a polynomial $g(x)$, and $h(x) = h_0 + \cdots + h_k x^k = \frac{x^n-1}{g(x)}$. It is not hard to see that the matrix

$$\mathbf{H} = \begin{pmatrix} 0 & 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \\ 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 & 0 \\ & & \ddots & & \ddots & & & \\ h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 & 0 \end{pmatrix}$$

satisfies the equation $\mathbf{H} \cdot \mathbf{G}^T = 0$. In other words, $\mathbf{H}$ is a parity-check matrix of the code $\mathcal{C}$. We have the following definition.

Definition 2.7. *Let $\mathcal{C} = \big(g(x)\big)$ be an (n,k)-cyclic code over $\mathbb{F}_q$. Then, the polynomial $g(x)$ is called the* generator polynomial *of $\mathcal{C}$; the polynomial $h(x) = \frac{x^n-1}{g(x)}$ is called the* parity-check polynomial *of $\mathcal{C}$.*

To illustrate these notions, we give a simple example.

Example 4. *Let $n = 4$ and $q = 3$. In $\mathbb{F}_3$, we have the following factorization*

$$x^4 - 1 = (x-1)(x+1)(x^2+1).$$

The cyclic code $\mathcal{C}$ generated by $g(x) = 1+x^2$ has the following matrix as a generator matrix.

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

The code $\mathcal{C}$ has $h(x) = x^2 + 2$ as a parity-check polynomial, and thus its corresponding parity-check matrix is

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{pmatrix}.$$

In the following paragraph, we consider a generalization of cyclic codes, namely quasi-cyclic codes. We saw that a cyclic code $\mathcal{C}$ has the property that if a word $\mathbf{a} \in \mathcal{C}$, then all of its right-shift words are also in $\mathcal{C}$. Now, a word of length $2n$ can be thought of as consisting of two components, each of which is a word of length n. Thus, if $\mathbf{c}^{(0)} = (c_0, \ldots, c_{2n-1})$, then its two components are $\mathbf{c}_1 = (c_0, \ldots, c_{n-1})$ and $\mathbf{c}_2 = (c_n, \ldots, c_{2n-1})$. The right-shift words of $\mathbf{c}^{(0)}$ are $\mathbf{c}^{(0)}, \mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(n-1)}$, where $\mathbf{c}^{(i)} = \big(\mathbf{c}_1^{(i)}, \mathbf{c}_2^{(i)}\big)$, *i.e.,* each component is the i-th right-shift word of that component of $\mathbf{c}^{(0)}$. If a linear code $\mathcal{C}$ of length $2n$ has the property that $\mathbf{c} \in \mathcal{C}$ implies that all of its right-shift words are also in $\mathcal{C}$, then $\mathcal{C}$ is called a quasi-cyclic code of order 2. If the number 2 is replaced by a general positive integer ℓ, one gets the following definition.

Definition 2.8. *Let $\ell \in \mathbb{N}$ and $\mathcal{C}$ be a linear code of length ℓn over $\mathbb{F}_q$. If $\mathcal{C}$ is closed under the right-shift operation, then $\mathcal{C}$ is called a quasi-cyclic code of order ℓ.*

If we take $\ell = 1$, we get the definition of cyclic codes. In code-based cryptography, two types of quasi-cyclic codes are often used, namely, quasi-cyclic codes of order 2 and 3 whose parity-check matrices are of the following forms

$$\mathfrak{H}_2 = \begin{pmatrix} \mathbf{I}_n & | & \mathbf{H} \end{pmatrix} \quad \text{and} \quad \mathfrak{H}_3 = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathbf{H}_1 \\ \mathbf{0} & \mathbf{I}_n & \mathbf{H}_2 \end{pmatrix}, \tag{2.4}$$

respectively. Here, $\mathbf{H}, \mathbf{H}_1, \mathbf{H}_2$ are circulant matrices, *i.e.,* square matrices in which the next row is the right-shift of the previous row.

2.1.3 Goppa Codes

Definition 2.9. *Let $g(x)$ be a polynomial of degree at most $n-1$ over an extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ and $L = \{\alpha_0, \ldots, \alpha_{n-1}\}$ a set of n distinct elements of $\mathbb{F}_{q^m}$ which does not contain any root of $g(x)$. Define*

$$\Gamma(L, g) = \left\{ (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n \;\middle|\; \sum_{i=0}^{n-1} c_i g(\alpha_i)^{-1} \frac{g(x) - g(\alpha_i)}{x - \alpha_i} = 0 \right\}.$$

The set $\Gamma(L, g)$ is called the Goppa *code with* Goppa *polynomial $g(x)$.*

The Goppa $\Gamma(L, g)$ code has the following property.

Theorem 2.2. *Let t be the degree of the* Goppa *polynomial $g(x)$. Then, the dimension of $\Gamma(L, g)$ is at least $n - mt$ and its minimum distance is at least $t + 1$.*

Proof. See [LN94]. □

2.2 Rank Metric Codes

2.2.1 Rank Metric and Rank Codes

This section provides some basic facts on rank metric. Comparisons of notions in rank metric and those in Hamming metric are often made. Let m, n be positive integers and $\mathbb{F}_{q^m}$ a finite extension of degree m over $\mathbb{F}_q$. Furthermore, let $B = \{\beta_1, \ldots, \beta_m\}$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then, for each element a in the extension field $\mathbb{F}_{q^m}$, it can be uniquely represented as

$$a = a_1\beta_1 + \cdots + a_m\beta_m,$$

with $a_1, \ldots, a_m$ belong to $\mathbb{F}_q$. By this way, a vector $\mathbf{a} = (a_1, \ldots, a_n)$ in the vector space $\mathbb{F}_{q^m}^n$ can be thought of as

$$\begin{pmatrix} a_1 \\ \vdots \\ a_m \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix},$$

where $\mathbf{A}$ is an $n \times m$ matrix with entries in $\mathbb{F}_q$. It is clear that when the basis B is fixed, then $\mathbf{a}$ and $\mathbf{A}$ correspond one-to-one to each other. As in Section 2.1, the HAMMING weight of $\mathbf{a}$ is the number of nonzero coordinates of $\mathbf{a}$, the rank weight of $\mathbf{a}$, however, is defined to be the rank of $\mathbf{A}$ over $\mathbb{F}_q$; it is denoted by $\operatorname{rank}(\mathbf{a})$. (When there is no ambiguity, it can also be denoted as $\|\mathbf{a}\|$, or $w(\mathbf{a})$.) A trivial observation shows that $\operatorname{rank}(\mathbf{a}) \leq \min\{m, n\}$ for all $\mathbf{a} \in \mathbb{F}_{q^m}^n$.

The distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ is the rank of $\mathbf{x} - \mathbf{y}$, or equivalently, of $\mathbf{X} - \mathbf{Y}$, and denoted by $d(\mathbf{x}, \mathbf{y})$. One has the following properties

Proposition 2.4. *Let $\mathbf{x}, \mathbf{y}, \mathbf{z}$ be vectors of $\mathbb{F}_{q^m}^n$. Then*

(*i*) $d(\mathbf{x}, \mathbf{y}) = 0$ *if and only if* $\mathbf{x} = \mathbf{y}$,

(*ii*) $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$,

(*iii*) $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$.

Proof. The first and second are straightforward. The thirds is in fact equivalent to the result

$$\operatorname{rank} \mathbf{A} + \operatorname{rank} \mathbf{B} \geq \operatorname{rank}(\mathbf{A} + \mathbf{B})$$

for any two matrices $\mathbf{A}$ and $\mathbf{B}$ of the same size. The last inequality comes from the fact that the image of $\mathbb{F}_q^m$ under $\mathbf{A} + \mathbf{B}$ is a subspace of the vector space formed by taking the sum of the images of $\mathbb{F}_q^m$ under each matrix. □

Another way to define the rank of a vector in $\mathbb{F}_{q^m}^n$ is through its coordinates, that is, the rank of $\mathbf{a}$ is the maximal number of its coordinates that are linearly independent over $\mathbb{F}_q$. One sees at once that this number is equal to the maximal number of rows of the matrix $\mathbf{A}$ that are linearly independent over $\mathbb{F}_q$, and thus the two ways are equivalent. Moreover, the maximal number of linearly independent coordinates of $\mathbf{a}$ is obviously less than or equal to the number of nonzero coordinates of $\mathbf{a}$, and thus, the rank weight of $\mathbf{a}$ is at most the HAMMING weight of $\mathbf{a}$. The $\mathbb{F}_q$-subspace of $\mathbb{F}_{q^m}$ (viewed as an m-dimensional vector space over $\mathbb{F}_q$) spanned by the coordinates of $\mathbf{a}$ is called the *support* of $\mathbf{a}$. Thus, $\operatorname{rank} \mathbf{a}$ is equal to the $\mathbb{F}_q$-dimension of the support of $\mathbf{a}$.

Example 5. *Let $q = 3, m = 3, n = 5$, and α a root of the polynomial $x^3 - x - 1$. We have $\mathbb{F}_3[\alpha] = \mathbb{F}_{3^3}$, and let $\{1, \alpha, \alpha^2\}$ be a basis of $\mathbb{F}_{3^3}$ over $\mathbb{F}_3$. The word $\mathbf{x} = (1, \alpha + 2, 0, \alpha^2 + 1, \alpha)$ has rank weight 3 since $1, \alpha + 2, \alpha^2 + 1$ are linearly independent over $\mathbb{F}_3$. On the other hand, its* Hamming *weight is 4. In terms of matrix, one has*

$$\begin{pmatrix} 1 \\ \alpha + 2 \\ 0 \\ \alpha^2 + 1 \\ \alpha \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \alpha \\ \alpha^2 \end{pmatrix}.$$

And indeed, the matrix $\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ *has rank equal to 3, since the 1st, 4th and 5th rows are linearly independent over $\mathbb{F}_3$.*

Similar to Hamming linear codes, a rank linear code can be defined by a generator matrix or a parity-check matrix.

Definition 2.10. *Let $\mathbf{G}$ be a $k \times n$ matrix of rank k with entries in $\mathbb{F}_{q^m}$. The set*

$$\mathcal{C} = \left\{ \mathbf{xG} \mid \mathbf{x} \in \mathbb{F}_{q^m}^k \right\}$$

is a (rank) linear code of length n and dimension k.

If $\mathbf{H}$ is a dual matrix of $\mathbf{G}$, then one has

Definition 2.11. *Let $\mathbf{H}$ be an $(n - k) \times n$ matrix of rank $n - k$ with entries in $\mathbb{F}_{q^m}$. Then*

$$\mathcal{C} = \left\{ \mathbf{c} \in \mathbb{F}_{q^m}^n \mid \mathbf{Hc}^T = \mathbf{0}^T \right\}$$

is an (n, k)-rank linear code.

Definition 2.12. *Let $\mathcal{C}$ be an (n, k)-rank linear code over $\mathbb{F}_{q^m}$. The minimum distance of $\mathcal{C}$ is defined by*

$$d_{\mathcal{C}} = \min \left\{ w(\mathbf{c}) \mid \mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\} \right\}.$$

From the above observation, one can deduce that $d_{\mathcal{C}} \leq n - k + 1$.

In the (binary) Hamming case, the number of words of weight k is $\binom{n}{k}$, represented by a binomial coefficient. In the rank case, we have an analogue to binomial coefficient, that is, Gaussian coefficient. For $k \leq n$, the Gaussian coefficient $\begin{bmatrix} n \\ k \end{bmatrix}_q$ is defined as follows

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \frac{(q^n - 1) \cdots (q^n - q^{k-1})}{(q^k - 1) \cdots (q^k - q^{k-1})},$$

and $\begin{bmatrix} n \\ 0 \end{bmatrix}_q = 1$. (Recall that $\binom{n}{k} = \frac{n\cdot(n-1)\cdots(n-k+1)}{k\cdot(k-1)\cdots 1}$.) The GAUSSian coefficient $\begin{bmatrix} n \\ k \end{bmatrix}_q$ represents the number of different subspcaces of dimension k of $\mathbb{F}_{q^n}$. Indeed, there are $(q^n-1)\cdots(q^n-q^{k-1})$ ways of choosing k linearly independent elements of $\mathbb{F}_{q^n}$, and in each space of dimension k, there are $(q^k-1)\cdots(q^k-q^{k-1})$ ways of choosing a basis. Thus, the number of different subspaces of dimension k is $\begin{bmatrix} n \\ k \end{bmatrix}_q$.

From the above result, one can deduce that the number of vectors of rank k in $\mathbb{F}_{q^m}^n$ is

$$T = (q^n-1)\cdots(q^n-q^{k-1})\begin{bmatrix} m \\ k \end{bmatrix}_q.$$

This can be done in this way. Let V be a subspace of $\mathbb{F}_{q^m}$ of dimension k, and $\{\alpha_1,\ldots,\alpha_k\}$ a basis for V over $\mathbb{F}_q$. Then, each vector $\mathbf{x}$, whose coordinates are in V, can be represented as

$$\mathbf{x} = (\alpha_1,\ldots,\alpha_k)\cdot\mathbf{M},$$

where $\mathbf{M}$ is a full rank $k\times n$ matrix over $\mathbb{F}_q$. It is not hard to see that the number of such matrices is $(q^n-1)\cdots(q^n-q^{k-1})$. From this, the statement follows. The above arguments can be stated in the following proposition.

Proposition 2.5. *Let k be a positive number such that $k \le \min\{m,n\}$, and*

$$\mathcal{S}_k^{n,m} = \left\{\mathbf{x}\in\mathbb{F}_{q^m}^n \mid \operatorname{rank}\mathbf{x} = k\right\}$$

the sphere of radius k centered at $\mathbf{0}$. Then $|\mathcal{S}_k^{n,m}| = (q^n-1)\cdots(q^n-q^{k-1})\begin{bmatrix} m \\ k \end{bmatrix}_q$.

The formula of $|\mathcal{S}_k^{n,m}|$ can be bounded as follows.

Proposition 2.6 ([Loi06])**.** *Let k be a positive number such that $k \le \min\{m,n\}$. Then,*

$$q^{k(m+n-k-2)} \le |\mathcal{S}_k^{n,m}| \le q^{k(m+n-k+1)}.$$

In most cases, the above result can somewhat be improved. Indeed, one has

$$\left(1-\frac{1}{q^{N-k+1}}\right)^k \le \left(1-\frac{1}{q^{N-k+1}}\right)\cdots\left(1-\frac{1}{q^N}\right) \le 1$$

for all integer $N \ge k$. Letting $N = n, m$ and multiplying the obtained inequalities, it turns out that

$$\left(1-\frac{1}{q^{n-k+1}}\right)^k\left(1-\frac{1}{q^{m-k+1}}\right)^k \le \prod_{i=0}^{k-1}\frac{\left(1-\frac{1}{q^{n-i}}\right)\left(1-\frac{1}{q^{m-i}}\right)}{\left(1-\frac{1}{q^{k-i}}\right)} \le \left(1-\frac{1}{q}\right)^{-k}.$$

The upper-bound can be rewritten as $b_1 = q^{k\left(1-\log_q(q-1)\right)}$, and the lower-bound $b_0 = q^{k\left(\log_q(q^{n-k+1}-1)+\log_q(q^{m-k+1}-1)+2k-m-n-2\right)}$. Thus, one has the following proposition.

Proposition 2.7. *Let k be a positive number such that $k \leq \min\{m, n\}$. Then,*

$$q^{k\left(\log_q(q^{n-k+1}-1)+\log_q(q^{m-k+1}-1)+k-2\right)} \leq |\mathcal{S}_k^{n,m}| \leq q^{k\left(m+n-k+1-\log_q(q-1)\right)}.$$

This section is concluded with the following proposition.

Proposition 2.8. *Let $\ell, m, n, w_1, \ldots, w_\ell$ be positive integers such that $m, n > d$, where $d = w_1 + \cdots + w_\ell$. Let $\mathbf{t}_i$ be randomly chosen from $\mathcal{S}_{w_i}^{n,m}$ for $i = 1, \ldots, \ell$, and $U = \mathrm{Supp}\left(\sum_{i=1}^{\ell} \mathbf{t}_i\right)$. Then, we have*

$$\Pr[\dim U = d] \geq 1 - \frac{1}{q^{m-d}} - \frac{1}{q^{n-d}}.$$

Proof. Let $\{\alpha_1, \ldots, \alpha_m\}$ be an arbitrary basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. For $i = 1, \ldots, \ell$, let $\{\beta_1^i, \ldots, \beta_{w_i}^i\}$ be a basis for $U_i = \mathrm{Supp}(\mathbf{t}_i)$ over $\mathbb{F}_q$, and $\mathbf{A}_i \in \mathbb{F}_q^{w_i \times m}$ such that

$$\begin{pmatrix} \beta_1^i \\ \vdots \\ \beta_{w_i}^i \end{pmatrix} = \mathbf{A}_i \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}.$$

Assume that for $i = 1, \ldots, \ell$, one has $\mathbf{t}_i = (t_{i,1}, \ldots, t_{i,n})$ and $t_{i,j} = \mathbf{v}_{i,j} \cdot (\beta_1^i, \ldots, \beta_{w_i}^i)^T$, where $\mathbf{v}_{i,j} \in \mathbb{F}_q^{w_i}$. For $j = 1, \ldots, n$, form the vector $\mathbf{v}_j = (\mathbf{v}_{1,j}, \ldots, \mathbf{v}_{\ell,j}) \in \mathbb{F}_q^d$, and let $V = \mathrm{Span}_{\mathbb{F}_q}(\mathbf{v}_1, \ldots, \mathbf{v}_n)$. Let $\mathbf{A} = [\mathbf{A}_1^T | \cdots | \mathbf{A}_\ell^T]^T \in \mathbb{F}_q^{d \times m}$, then

$$U = \left\{ \mathbf{v} \cdot \mathbf{A} \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix} \;\middle|\; \mathbf{v} \in V \right\}.$$

Let X be the event that a randomly chosen matrix $\mathbf{A}$ satisfies $\mathrm{rank}\,\mathbf{A} = d$, and Y the event that randomly chosen matrices $\mathbf{A}_i$'s satisfy $\mathrm{rank}\,\mathbf{A}_i = w_i$ for all $i = 1, \ldots, \ell$. By Bayes's formula, we have

$$\begin{aligned} \Pr[X|Y] &= \frac{\Pr[Y|X] \cdot \Pr[X]}{\Pr[Y]} \\ &= \frac{\Pr[X]}{\Pr[Y]} \\ &= \frac{\Pr[\mathrm{rank}\,\mathbf{A} = d \mid \mathbf{A} \leftarrow \mathbb{F}_q^{d \times m}]}{\prod_{i=1}^{\ell} \Pr[\mathrm{rank}\,\mathbf{A}_i = w_i \mid \mathbf{A}_i \leftarrow \mathbb{F}_q^{w_i \times m}]}. \end{aligned}$$

We need the following lemma.

Lemma 2. *Let $t \leq m$ be positive integers and $\mathbf{M}$ a random matrix in $\mathbb{F}_q^{t\times m}$. Then*

$$\Pr[\operatorname{rank}\mathbf{M} = t] = \left(1 - \frac{1}{q^m}\right)\cdots\left(1 - \frac{1}{q^{m-t+1}}\right).$$

Proof. There are $q^m - 1$ possibilities for the choice of the first row, $q^m - q$ for the second, ..., and $q^m - q^{t-1}$ for the tth row. Thus the number of matrices of full rank makes up to $(q^m - 1)\cdots(q^m - q^{t-1})$. Therefore, one has

$$\begin{aligned}\Pr[\operatorname{rank}\mathbf{M} = t \mid \mathbf{M} \leftarrow \mathbb{F}_q^{t\times m}] &= \frac{(q^m - 1)\cdots(q^m - q^{t-1})}{q^{tm}} \\ &= \left(1 - \frac{1}{q^m}\right)\cdots\left(1 - \frac{1}{q^{m-t+1}}\right).\end{aligned}$$

□

From Lemma 2, one easily sees that

$$\begin{aligned}\Pr[X|Y] &\geq \Pr[\operatorname{rank}\mathbf{A} = d \mid \mathbf{A} \leftarrow \mathbb{F}_q^{d\times m}] \\ &\geq 1 - \frac{1}{q^{m-d+1}} - \cdots - \frac{1}{q^m} \\ &\geq 1 - \frac{1}{q^{m-d}}.\end{aligned}$$

By the same argument, one obtains

$$\Pr[V = \mathbb{F}_q^d] \geq 1 - \frac{1}{q^{n-d}}.$$

Let

$$\mathfrak{U} = \{\mathbf{v}\cdot\mathbf{A} \mid \mathbf{v} \in V\}.$$

The event $\dim U = d$ is equivalent to the event $\dim \mathfrak{U} = d$, and thus equivalent to the event $\left(V = \mathbb{F}_q^d \cap \operatorname{rank}\mathbf{A} = d\right)$. By the independence of $\mathbf{t}_i$'s, we see that the two events $V = \mathbb{F}_q^d$ and $\operatorname{rank}\mathbf{A} = d$ are independent. Therefore,

$$\begin{aligned}\Pr[\dim U = d] &= \Pr[V = \mathbb{F}_q^d]\cdot\Pr[\operatorname{rank}\mathbf{A} = d] \\ &\geq \left(1 - \frac{1}{q^{n-d}}\right)\cdot\left(1 - \frac{1}{q^{m-d}}\right) \\ &\geq 1 - \frac{1}{q^{m-d}} - \frac{1}{q^{n-d}}.\end{aligned}$$

□

2.2.2 Gabidulin Codes

In this section, we implicitly assume that $n \leq m$.

Definition 2.13. *Let* $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{F}_{q^m}^n$ *be a vector of rank* n*. The* (n, k) GABIDULIN *code* $\mathcal{G}_{\mathbf{g},k}$ *of support* $\mathbf{g}$ *is the linear code generated by the matrix*

$$\mathbf{G}_{\mathbf{g},k} = \begin{pmatrix} g_1 & g_2 & \cdots & g_n \\ g_1^q & g_2^q & \cdots & g_n^q \\ \vdots & \vdots & \vdots & \vdots \\ g_1^{q^{k-1}} & g_2^{q^{k-1}} & \cdots & g_n^{q^{k-1}} \end{pmatrix}.$$

Connected to this matrix is the notion of q-polynomials.

Definition 2.14. *Let* d *be a non-negative integer. A* q*-polynomial of* q*-degree* d *over* $\mathbb{F}_{q^m}$ *is a polynomial of the form*

$$f(x) = a_0 x + a_1 x^q + \cdots + a_d x^{q^d},$$

where $a_0, \ldots, a_d \in \mathbb{F}_{q^m}$ *and* $a_d \neq 0$*. The* q*-degree of* $f(x)$ *is denoted by* $\deg_q(f)$*.*

One can verify that for any $\alpha, \beta \in \mathbb{F}_{q^m}, a \in \mathbb{F}_q$, one has $f(a\alpha) = af(\alpha)$ and $f(\alpha + \beta) = f(\alpha) + f(\beta)$. [2] Thus, q-polynomials are also known as *linearized* polynomials. In the language of q-polynomials, GABIDULIN codes can be interpreted as follows.

Definition 2.15. *Let* $\mathbf{g} = (g_1, \ldots, g_n) \in \mathbb{F}_{q^m}^n$ *be a vector of rank* n*. The* (n, k) GABIDULIN *code of support* $\mathbf{g}$ *is the set*

$$\mathcal{G}_{\mathbf{g},k} = \left\{ \big(f(g_1), \ldots, f(g_n)\big) \mid f(x) \in \mathbb{F}_{q^m}[x], \deg_q(f) \leq k-1 \right\}.$$

The code $\mathcal{G}_{\mathbf{g},k}$ has the following property.

Proposition 2.9. *Let* d_r *be the minimum (rank) distance of* $\mathcal{G}_{\mathbf{g},k}$*. Then,* $d_r = n - k + 1$*.*

Proof. Let $f(x)$ be a q-polynomial of q-degree $d \leq k-1$ and $\mathbf{c} = \big(f(g_1), \ldots, f(g_n)\big)$. We will show that $\operatorname{rank}(\mathbf{c}) \geq n - d$.

By the linearity of f, we can think of f as a linear map. Thus,

$$\operatorname{rank}(\mathbf{c}) = \dim \operatorname{Im}(f) \geq n - \dim \ker(f).$$

[2] These come from the fact that $a^q = a$ and $(\alpha + \beta)^q = \alpha^q + \beta^q$.

(The equality holds when $\ker(f) \subseteq \mathrm{Span}_{\mathbb{F}_q}(g_1, \ldots, g_n)$.) Since $\deg_q(f) = d$, so $\dim \ker(f) = d$. The inequality follows.

We have $\mathrm{rank}(\mathbf{c}) \geq n - d \geq n - k + 1$ for any $\mathbf{c} \in \mathcal{G}_{\mathbf{g},k}$, and thus $d_r \geq n - k + 1$. On the other hand, we already have $d_r \leq n - k + 1$. Therefore, one must have $d_r = n - k + 1$. □

2.3 Modern Cryptography

This section covers the very basic notions of modern cryptography related directly to our works.

Notation. By writing $x \leftarrow X$, we mean that x is drawn according to the distribution X, if X is a distribution; or drawn uniformly at random from X when X is a set, or the output of the algorithm X, if X is an algorithm.

2.3.1 The Computational Model

Definition 2.16 (Asymptotic functions)**.** *Let $f, g \colon \mathbb{N} \to \mathbb{N}$ be two functions. Then*

- *$g(n) = O(f(n))$ if there exists a constant $c > 0$ such that $g(n) \leq c \cdot f(n)$ for every large enough n.*
- *$g(n) = o(f(n))$ if for every $c > 0$, $g(n) \leq c \cdot f(n)$ for every large enough n.*
- *$g(n) = \Omega(f(n))$ if $f(n) = O(g(n))$,* i.e., *there exists a constant $c > 0$ such that $g(n) \geq c \cdot f(n)$ for every large enough n.*
- *$g(n) = \omega(f(n))$ if $f(n) = o(g(n))$,* i.e., *for every $c > 0$, $g(n) \geq c \cdot f(n)$ for every large enough n.*

Negligible Functions. A function $f \colon \mathbb{N} \to \mathbb{R}$ is negligible if for every polynomial $p(x)$, there exists a positive integer n_0 such that for all $n > n_0$, it holds that

$$f(n) \leq \frac{1}{p(n)}.$$

Example 6. *1. The function $f(x) = 2^{-n}$ is a negligible function.*

2. Any polynomial is not a negligible function.

Now, let D be a distribution over a countable set Ω. For an element $x \in \Omega$, the notation $D(x)$ signifies the probability that an element of Ω chosen according to D is equal to x.

Statistical Distances. Let D_1 and D_2 be two distributions over Ω. The *statistical distance* between D_1 and D_2 is defined as

$$\Delta(D_1, D_2) = \frac{1}{2} \sum_{x \in \Omega} \big| D_1(x) - D_2(x) \big|.$$

Two distributions are said to be *statistically indistinguishable* if there exists a negligible function $\mu(\cdot)$ (in the security parameter n) such that

$$\Delta(D_1, D_2) \leq \mu(n).$$

In terms of distinguisher, *i.e.,* a probabilistic polynomial-time algorithm whose output is either 0 or 1, statistical distance can be stated as follows.

Definition 2.17. *The statistical distance between D_1 and D_2 is*

$$\Delta(D_1, D_2) = \max_A \Big| \Pr_{x \leftarrow D_1}[A(x) = 1] - \Pr_{y \leftarrow D_2}[A(y) = 1] \Big|,$$

where the maximum is taken over all possible distinguishers A.

Computational Distances. Let $\{x_n\}_{n \in \mathbb{N}}$ and $\{y_n\}_{n \in \mathbb{N}}$ be a two sequences of random variables and A a probabilistic polynomial-time algorithm that either outputs 0 or 1. The *computational distance* between $\{x_n\}_{n \in \mathbb{N}}$ and $\{y_n\}_{n \in \mathbb{N}}$ is defined to be the quantity

$$\big| \Pr[A(x_n) = 1] - \Pr[A(y_n) = 1] \big|.$$

As similar as statistically indistinguishable, two sequences of random variables are said to be *computationally indistinguishable* if there exists a negligible function $\mu(\cdot)$ such that

$$\big| \Pr[A(x_n) = 1] - \Pr[A(y_n) = 1] \big| \leq \mu(n).$$

Intuitively, the above definition can be understood in a way that any probabilistic polynomial-time algorithm outputs 1 with almost the same probability whether its input is from the first or the second sequence.

2.3.2 Public-key Encryption

Suppose that Bob and Alice want to exchange messages through a public channel, say the Internet. Since it is a public channel, so anyone could see the exchanged messages. This undesirable property compels both of them to encrypt their messages before sending. Of cause, the kind of encryption should allow each of them to be able to read other's messages. One way to accomplish this task is that Bob and Alice agree with each other beforehand on a single key which is used to encrypt as well as decrypt. This kind of encryption is called symmetric encryption.

Another solution is that Alice generates a pair consisting of a public key pk and a secret key sk. The key pk is sent to Bob regardless of the fact that others can learn about it. Bob uses this key to encrypt his message while Alice uses her secret key sk to decrypt Bob's encrypted message. As for Bob, he also generates his own key pair, and does as Alice. This kind of encryption is called asymmetric encryption or public-key encryption. To provide the basic definition and security requirement of a public-key encryption scheme is the aim of this section.

An intuitive figure of public-key encryption is already described in the above paragraph, its formal definition is as follows.

Definition 2.18. *A public-key encryption (PKE) scheme $\mathcal{E}$ is a tuple $\mathcal{E} = (\mathsf{Set}, \mathsf{Enc}, \mathsf{Dec})$ of three algorithms:*

- $\mathsf{Set}(1^\lambda)$ *outputs public and secret keys* $(\mathsf{pk}, \mathsf{sk})$ *for a security parameter* λ,
- $\mathsf{Enc}(\mathsf{pk}, m)$ *on input public key* pk *and a message* $m \in \mathcal{M}$, *the allowed message space, outputs ciphertext* ct,
- $\mathsf{Dec}(\mathsf{sk}, ct)$ *on input secret key* sk *and ciphertext* ct, *outputs messages* m'.

It is required that for any $m \in \mathcal{M}$,

$$\Pr_{m \in \mathcal{M}}\big[\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) \neq m\big] \leq \mathsf{negl}(\lambda).$$

The security of a public key scheme is usually analyzed with the chosen plaintext attack (CPA) experiment (or game).

The CPA indistinguishability experiment $\mathsf{Pub}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$:

1. $\mathsf{Gen}(1^n)$ is run to obtain keys $(\mathsf{pk}, \mathsf{sk})$.
2. Adversary $\mathcal{A}$ is given pk as well as ability to access to $\mathsf{Enc}(\mathsf{pk}, \cdot)$, and outputs a pair of messages m_0, m_1 of the same length.
3. A random bit $b \leftarrow \{0, 1\}$ is chosen, and then a ciphertext $c \leftarrow \mathsf{Enc}(\mathsf{pk}, m_b)$ is computed and given to $\mathcal{A}$. Ciphertext c is called the challenge ciphertext.
4. $\mathcal{A}$ continues to have access to $\mathsf{Enc}(\mathsf{pk}, \cdot)$, and finally, outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise.

Definition 2.19. *A public-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has* indistinguishable encryption under a chosen-plaintext attack *(or is* CPA secure*) if for all probabilistic polynomial-time adversaries $\mathcal{A}$, there exists a negligible function* negl *such that*

$$\Pr[\mathsf{Pub}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n).$$

2.3.3 Zero-Knowledge Proof Systems

This section provides some basic notions on zero-knowledge proofs. The main purpose is to give an intuitive idea of these proof systems, which will serve for later sections and chapters. Most of the notions are also taken from [Gol01].

Throughout this section, P and V will stand for interactive machines, and on common input x, the notion $\langle P, V\rangle(x)$ denotes the (probabilistic) output of V after interacting with P.

2.3.3.1 Interactive Proofs

Definition 2.20. *Let $L \subseteq \{0,1\}^*$ be a language. A pair of interactive machines (P, V) is called* an interactive proof system *for L if V is a probabilistic polynomial-time machine and the following properties hold.*

(i) Completeness. *For every $x \in L$,*

$$\Pr[\langle P, V\rangle(x) = 1] \geq \frac{2}{3}.$$

(ii) Soundness. *For every $x \notin L$ and any interactive machine P^*,*

$$\Pr[\langle P^*, V\rangle(x) = 1] \leq \frac{1}{3}.$$

The second property says that a cheating prover can fool a verifier with probability at most $\frac{1}{3}$. By repeating the protocol many times, this probability can be made negligible. Thus, a cheating prover essentially cannot fool a verifier to accept an input that is not in L.

2.3.3.2 Computationally Sound Proofs

In Definition 2.20, if we restrict the power of the prover to be polynomial-time and relax the soundness property to be *infeasible* to fool the verifier, we get the notion of computationally sound proofs.

Definition 2.21 (Computationally Sound Proof System)**.** *A pair of interactive machines (P, V) is called a* computationally sound proof system *(or an argument) for a language L if both machines are polynomial-time (with auxiliary inputs) and the following conditions hold.*

- Completeness. *For any $x \in L$, there exists a string y such that for every string z (as auxiliary input of V),*

$$\Pr[\langle P(y), V(z)\rangle(x) = 1] \geq \frac{2}{3}.$$

- Computational soundness. *For every polynomial-time machine P^* and for all sufficiently long $x \notin L$, and any y, z,*

$$\Pr[\langle P^*(y), V(z)\rangle(x) = 1] \leq \frac{1}{3}.$$

2.3.3.3 Zero-Knowledge Proofs

Zero-knowledge proofs were first introduced by Goldwasser *et al.* [GMR85]. Zero-knowledge is an additional property of the prover P of an interactive system (P, V). Intuitively, this property can be understood in a way that the interaction with P does not help V to be more efficient in any computing task related to the common input x (of the interaction), or whatever can be computed efficiently after the interaction, V can also do it without interacting with P. This notion is captured by simulation paradigm. It is as follows.

Definition 2.22. *Let $L \subseteq \{0,1\}^*$ and (P, V) an interactive proof system for L. The prover P is said to be perfect zero-knowledge if for every probabilistic polynomial-time interactive machine V^*, there exists a probabilistic polynomial-time algorithm S^* such that for any $x \in L$, the following conditions hold.*

1. *Machine S^* outputs a special symbol, denoted by $\perp$, with probability at most $\frac{1}{2}$.*
2. *Let $s^*(x)$ denote the variable describing the output of S^* conditioned on $S^*(x) \neq \perp$. Then the following variables are identically distributed.*
   - *$\langle P, V^*\rangle(x)$,* i.e., *the output of V^* after interacting with P on common input x.*
   - *$S^*(x)$,* i.e., *the output of algorithm S^* on input x.*

S^* is called a simulator *for the interaction between P and V^*.*

In the above definition, if the condition identical is relaxed by *statistically close* or even weaker by *computationally close*, we get the notion of *statistical* zero-knowledge and *computational* zero-knowledge, respectively.

2.3.3.4 Proofs of Knowledge

Definition 2.23. *Let R be a binary relation and $\kappa\colon \mathbb{N} \to [0,1]$ a function. An interactive machine V is called a knowledge verifier for R with knowledge error κ if the following conditions hold:*

- Non-triviality. *There exists an interactive machine P such that for every $(x, y) \in R$, every possible interaction of V with P on the pair (x, y) are accepted.*

- Validity. *There exist a polynomial $q(\cdot)$ and a probabilistic machine K such that for every interactive machine P, every $x \in L_R$, and every $y, r \in \{0,1\}^*$, machine K satisfies the condition: Denote by $p(x, y, r)$ the probability that V accepts when interacting with the prover specified by $P_{x,y,r}$,* i.e., *the function describes messages sent by P on input x, auxiliary input y, and random input r. If $p(x, y, r) > \kappa(|x|)$, then, on input x and with access to $P_{x,y,r}$, machine K outputs a solution $s \in R(x)$ with an expected number of steps bounded by $q(x, y, r)/\big(p(x, y, r) - \kappa(|x|)\big)$. (The machine K is called a knowledge extractor.) An interactive pair (P, V) in which V is a knowledge verifier for a relation R and P satisfies the non-triviality condition is called a system for proofs of knowledge for the relation R.*

2.4 Hardness Assumptions

This section recall some problems which are widely believed to be inefficient to solve. These problems are parted in the Hamming metric class and rank metric class with respect to the involved metric.

2.4.1 Hamming Metric Problems

In 1978, Berlekamp *et al.* [BMvT78] showed that the problem of decoding linear codes and the problem of finding small weight codewords of a linear code are both $\mathcal{NP}$-complete. This result suggests that this problem can be used for cryptography design. The general decoding problem is as follows.

Problem 1 (Computational Syndrome Decoding Problem)**.** *Let $\mathbf{H}$ be a matrix in $\mathbb{F}_q^{(n-k)\times n}$, $\mathbf{y}$ a word in $\mathbb{F}_q^n$, and w a positive integer. Find (if any) a word $\mathbf{x}$ of* Hamming *weight w such that $\mathbf{H}\mathbf{x}^T = \mathbf{y}^T$?*

For abbreviation, this problem is traditionally denoted by $\mathsf{CSD}(n, k, w)$ or $\mathsf{CSD}(\mathbf{H}, \mathbf{y}, w)$. The problem of finding small weight codewords of a linear code is a specialization of Problem 1 by letting $\mathbf{y} = \mathbf{0}$. It is stated as follows.

Problem 2. *Let $\mathbf{H}$ be a matrix in $\mathbb{F}_q^{(n-k)\times n}$ and w a positive integer. Find (if any) a word $\mathbf{x}$ of* Hamming *weight w such that $\mathbf{H}\mathbf{x}^T = \mathbf{0}^T$?*

For the designing of cryptosystems, the decisional version of Problem 1 is usually used.

Definition 2.24 (Syndrome Decoding Distribution)**.** *Let n, k, and w be positive integers. The syndrome decoding distribution, denoted by $\mathsf{SD}(n, k, w)$, chooses $\mathbf{H} \leftarrow \mathbb{F}_q^{(n-k)\times n}$ and $\mathbf{x} \leftarrow \mathcal{S}_w^n$, and outputs $\left(\mathbf{H}, \mathbf{H} \cdot \mathbf{x}^T\right)$.*

Problem 3 (Decisional Syndrome Decoding Problem)**.** *Let $(\mathbf{H}, \mathbf{y})$ be an instance either from the $\mathsf{SD}(n, k, w)$ distribution or the uniform distribution over $\mathbb{F}_q^{(n-k)\times n} \times \mathbb{F}_q^n$. Decide which is the case?*

Problem 1 and Problem 3 are fundamental elements for designing code-based cryptography. The principal attack on Problem 1 and 2 is the information set decoding algorithm which was initiated by Prange [Pra62] and then Stern [Ste89]. Recent developments of these algorithms could be found in [BLP11] and [BJMM12]. In some cases, in which matrix $\mathbf{H}$ has specific structures such as it is a parity-check matrix of a Goppa code or of the form $\mathfrak{H}_2$ or $\mathfrak{H}_3$ as in Equation 2.4, these two problems are also (heuristically) considered to be inefficient to solve.

2.4.2 Rank Metric Problems

By changing the metric and fixing attention on a finite extension $\mathbb{F}_{q^m}$ instead of $\mathbb{F}_q$, Problem 1 changes itself in the following problem:

Problem 4 (Rank Syndrome Decoding Problem)**.** *Let $\mathbf{H}$ be a matrix in $\mathbb{F}_{q^m}^{(n-k)\times n}$, $\mathbf{y}$ a word in $\mathbb{F}_{q^m}^n$, and w a positive integer. Find (if any) a word $\mathbf{x}$ of rank weight w such that $\mathbf{H}\mathbf{x}^T = \mathbf{y}^T$?*

In the literature, Problem 4 is usually referred to as RSD problem, or more specifically, $\mathsf{RSD}(n, k, w)$. The hardness of the above problem is *probabilistically* reduced to that of Problem 1 [GZ16]. In Definition 2.24, if Hamming metric is replaced by rank metric, one gets the rank syndrome decoding distribution. The distinguishing problem for rank metric is stated as follows.

Problem 5 (Decisional Rank Syndrome Decoding Problem)**.** *Let $(\mathbf{H}, \mathbf{y})$ be an instance either from the $\mathsf{RSD}(n, k, w)$ distribution or the uniform distribution over $\mathbb{F}_{q^m}^{(n-k)\times n} \times \mathbb{F}_{q^m}^n$. Decide which is the case?*

There are two main methods for efficiently solving Problem 4, namely, the combinatoric attacks [GRS16, AGHT18] and the algebraic attack [BBC+20]. The matrix $\mathbf{H}$ in Problem 4 and Problem 5 is completely random and has no structure. These problems in the cases which $\mathbf{H}$ takes on the quasi-cyclic or the ideal form are also consider to be hard. The ideal form of a matrix is a generalization of the quasi-cyclic form and will be defined in Section 4.2.2.

2.5 Code-based Cryptosystems

This section presents cryptosystems based on codes, some of which are directly related to our works. These cryptosystems are McEliece's cryptosystem, Stern's identification protocol in both Hamming and rank metric, and the Hamming quasi-cyclic (HQC) encryption scheme.

2.5.1 McEliece's Cryptosystem

This cryptosystem was introduced by McEliece in 1978 [McE78].

1. **Key Generation.** This algorithm performs the following steps:
   - Select a matrix $\mathbf{G}$ of an $[n, k]$-Goppa code. This code can decode up to t errors.
   - Randomly choose an invertible matrix $\mathbf{S}$ of size $k \times k$ and a permutation matrix $\mathbf{P}$ of size $n \times n$ over $\mathbb{F}_q$.
   - Compute $\widehat{\mathbf{G}} = \mathbf{S} \cdot \mathbf{G} \cdot \mathbf{P}$.
   - Output $\mathsf{pk} = (\widehat{\mathbf{G}}, t)$ and $\mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P})$.
2. **Encryption.** To encrypt a message $\mathbf{m}$, choose a random vector $\mathbf{e}$ of weight at most t and compute the ciphertext
$$\mathbf{c} = \mathbf{m}\widehat{\mathbf{G}} + \mathbf{e}.$$
3. **Decryption.** A ciphertext $\mathbf{c}$ is decrypted in the following way:
   - First, the matrix $\mathbf{P}$ is used to compute
$$\mathbf{c} \cdot \mathbf{P}^{-1} = \mathbf{mS} \cdot \mathbf{G} + \mathbf{e} \cdot \mathbf{P}^{-1}.$$
   - Next, the decoding algorithm of the Goppa code $\mathcal{C}$ is used to recover $\mathbf{m} \cdot \mathbf{S}$.
   - Finally, by right-multiplying with $\mathbf{S}^{-1}$, the message $\mathbf{m}$ is recovered.

2.5.2 Stern Identification Protocol

In 1994, Jacques Stern introduced an identification scheme based on codes [Ste94]. The underlying hard problem is the syndrome decoding problem. The scheme is an interactive 3-move protocol between a prover and a verifier. The prover $\mathcal{P}$, who possesses a piece of secret information, tries to convince the verifier

$\mathcal{V}$ of that possession without showing the secret itself. The common inputs to both parties consist of a random binary matrix $\mathbf{H}$ of size $(n-k)\times n$ and a syndrome $\mathbf{s}\in\mathbb{F}_2^{n-k}$. The secret information of the prover is a vector $\mathbf{x}$ in $\mathbb{F}_2^n$ of small HAMMING weight t satisfying that $\mathbf{H}\mathbf{x}^T=\mathbf{s}^T$. The scheme is as follows.

1. **Commitment.** The prover $\mathcal{P}$ randomly chooses a vector $\mathbf{y}\leftarrow\mathbb{F}_2^n$ and a permutation $\sigma\leftarrow S_n$. He computes

$$\begin{cases} c_1 = h\big(\sigma, \mathbf{H}\mathbf{y}^T\big), \\ c_2 = h\big(\sigma(\mathbf{y})\big), \\ c_3 = h\big(\sigma(\mathbf{x}+\mathbf{y})\big), \end{cases}$$

   where h is a public hash function. Then, $\mathcal{P}$ sends $cmt=(c_1,c_2,c_3)$ to the verifier $\mathcal{V}$.

2. **Challenge.** Upon receiving cmt, $\mathcal{V}$ randomly picks a bit $b\leftarrow\{1,2,3\}$ and sends it to $\mathcal{P}$.

3. **Response.** $\mathcal{P}$ responses as follows.
   (a) If $b=1$, then $\mathcal{P}$ releases $\sigma(\mathbf{x})$ and $\sigma(\mathbf{y})$.
   (b) If $b=2$, then $\mathcal{P}$ releases σ and $\mathbf{x}+\mathbf{y}$.
   (c) If $b=3$, then $\mathcal{P}$ releases σ and $\mathbf{y}$.

4. **Verify.** $\mathcal{V}$ verifies as follows.
   (a) If $b=1$, then $\mathcal{V}$ checks for the validity of c_2 and c_3, and $w\big(\sigma(\mathbf{x})\big)=t$.
   (b) If $b=2$, then $\mathcal{V}$ checks for the validity of c_1 and c_3.
   (c) If $b=3$, then $\mathcal{V}$ checks for the validity of c_1 and c_2.

   If all the checks are correct, then $\mathcal{P}$ outputs 1; otherwise, it outputs 0.

2.5.3 HQC Scheme

The HQC cryptosystem was introduced by AGUILAR *et al.* [ABD+16] and has reached the 3rd round of NIST's call for post-quantum cryptography. The scheme makes use of two types of codes as defined in Definition 2.8 with $\ell=2,3$, *i.e.*, codes defined by $\tilde{\mathfrak{H}}_2$ and $\tilde{\mathfrak{H}}_3$. The multiplication of two vectors of length n is defined through the multiplication of two corresponding polynomials in the ring $\mathbb{F}_2[x]/(x^n-1)$ for a suitable n. That is,

$$\mathbf{x}\cdot\mathbf{y}=\phi^{-1}\big(\phi(\mathbf{x})\cdot\phi(\mathbf{y})\big),$$

where ϕ is the map described in Section 2.1.2 with $q = 2$. The structure of quasi-cyclic codes makes the scheme's public key size quite small. The detailed description of the scheme is as follows.

1. $\mathsf{HQC.Setup}(1^\lambda)$: Generate parameters $n = n(\lambda), k = k(\lambda), \delta = \delta(\lambda), w = w(\lambda), w_e = w_e(\lambda), w_r = w_r(\lambda)$. The plaintext space is $\mathbb{F}_2^k$. Output $\mathsf{param} = (n, k, \delta, w, w_e, w_r)$.

2. $\mathsf{HQC.KeyGen}(\mathsf{param})$: Generate $\mathbf{h} \leftarrow \mathbb{F}_2^n, \mathbf{x}, \mathbf{y} \leftarrow \mathcal{S}_w^n$, a generator matrix $\mathbf{G} \in \mathbb{F}_2^{k \times n}$ of a public code $\mathcal{C}$, which is capable of correcting up to δ errors. Output $\mathsf{pk_{HQC}} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}, \mathbf{G})$ and $\mathsf{sk_{HQC}} = (\mathbf{x}, \mathbf{y})$.

3. $\mathsf{HQC.Enc}(\mathsf{pk_{HQC}}, \mathbf{m})$: To encrypt a message $\mathbf{m} \in \mathbb{F}_2^k$, randomly choose $\mathbf{r}_1, \mathbf{r}_2 \leftarrow \mathcal{S}_{w_r}^n$ and $\mathbf{e} \leftarrow \mathcal{S}_{w_e}^n$ Compute

$$\begin{cases} \mathbf{c}_1 = \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2, \\ \mathbf{c}_2 = \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e} + \mathbf{m} \cdot \mathbf{G}. \end{cases}$$

   Return $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$.

4. $\mathsf{HQC.Dec}(\mathsf{sk_{HQC}}, \mathbf{c})$: Apply the decoding algorithm of the code $\mathcal{C}$ to

$$\mathbf{y} \cdot \mathbf{c}_1 - \mathbf{c}_2 = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{y} \cdot \mathbf{r}_1 + \mathbf{e} + \mathbf{m} \cdot \mathbf{G}.$$

Security of HQC scheme relies on the hardness assumptions concerning 2- and 3-quasi cyclic codes.

2.5.4 Rank Stern Identification Protocol

In 1995, Kefei Chen proposed an identification scheme in the rank metric context with a remarkable property that it does not make use of random oracles, *i.e.,* hash functions [Che96]. However, sixteen years later, this scheme was broken by the attacks of Gaborit *et al.* [GSZ11]. Also, an identification scheme was introduced in [GSZ11]. This scheme, in its essence, can be regarded as the rank version of the Stern's identification protocol. Despite of this fact, the scheme differentiates itself from the latter by the use of a new operation. It is defined in the next paragraph.

As in Section 2.2.1, let $B = \{\beta_1, \ldots, \beta_m\}$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then, each vector $\mathbf{x}$ can be uniquely mapped to a matrix $\mathbf{A_x}$ of the size $m \times n$ over $\mathbb{F}_q$. Let us call this map φ_B. For an invertible matrix $\mathbf{Q}$ of size m over $\mathbb{F}_q$, the operation $\mathbf{Q} \star \mathbf{x}$ is defined to be equal to the vector whose image under φ_B is $\mathbf{Q} \cdot \mathbf{A_x}$. This operation is formally stated in the following definition.

Definition 2.25. *Let* $\mathbf{Q}$ *be an element of* $\mathrm{GL}(m,q)$ *and* $B=\{\beta_1,\dots,\beta_m\}$ *a basis for* $\mathbb{F}_{q^m}$ *over* $\mathbb{F}_q$*. Let* $\varphi_B\colon \mathbb{F}_{q^m}^n \to \mathbb{F}_q^{m\times n}$ *be the map associating each vector to its representation matrix with respect to* B*. Then, for each* $\mathbf{x}\in\mathbb{F}_{q^m}^n$

$$\mathbf{Q}\star\mathbf{x}=\varphi_B^{-1}\big(\mathbf{Q}\cdot\varphi_B(\mathbf{x})\big).$$

This operation has the following properties.

Proposition 2.10. *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be two vectors in* $\mathbb{F}_{q^m}^n$*. Then*

(*i*) $\mathrm{rank}(\mathbf{x})=\mathrm{rank}(\mathbf{Q}\star\mathbf{x})$ *for any* $\mathbf{Q}\in\mathrm{GL}(m,q)$;

(*ii*) $(\mathbf{Q}\star\mathbf{x})\mathbf{P}=\mathbf{Q}\star(\mathbf{x}\mathbf{P})$ *for any* $\mathbf{Q}\in\mathrm{GL}(m,q)$ *and* $\mathbf{P}\in\mathrm{GL}(n,q)$;

(*iii*) *if* $\mathrm{rank}(\mathbf{x})=\mathrm{rank}(\mathbf{y})$*, then, there exist matrices* $\mathbf{Q}\in\mathrm{GL}(m,q)$ *and* $\mathbf{P}\in\mathrm{GL}(n,q)$ *such that* $\mathbf{y}=\mathbf{Q}\star\mathbf{x}\mathbf{P}$.

Proof. (*i*) By the definition, we have

$$\mathrm{rank}(\mathbf{Q}\star\mathbf{x})=\mathrm{rank}\left(\mathbf{Q}\cdot\varphi_B(\mathbf{x})\right).$$

Since $\mathbf{Q}$ is invertible, $\mathrm{rank}\left(\mathbf{Q}\cdot\varphi_B(\mathbf{x})\right)=\mathrm{rank}\left(\varphi_B(\mathbf{x})\right)=\mathrm{rank}(\mathbf{x})$.

(*ii*) Observe that $\varphi_B(\mathbf{x}\mathbf{P})=\varphi_B(\mathbf{x})\mathbf{P}$ for any $\mathbf{P}\in\mathrm{GL}(n,q)$. By applying the map φ_B to both $(\mathbf{Q}\star\mathbf{x})\mathbf{P}$ and $\mathbf{Q}\star(\mathbf{x}\mathbf{P})$, we get the same value. Thus $(\mathbf{Q}\star\mathbf{x})\mathbf{P}=\mathbf{Q}\star(\mathbf{x}\mathbf{P})$.

(*iii*) Let U and V be the supports of $\mathbf{x}$ and $\mathbf{y}$, respectively. Assume that $\mathrm{rank}(\mathbf{x})=\mathrm{rank}(\mathbf{y})=d$, so we have $\dim U=\dim V=d$. Let $\{e_1,\dots,e_d\}$ and $\{f_1,\dots,f_d\}$ be bases for U and V over $\mathbb{F}_q$, respectively. Here, e_i's and f_i are thought of as elements of $\mathbb{F}_q^m$.

Since U and V are subspaces of $\mathbb{F}_{q^m}$, so their bases can be extended to bases for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Assume that

$$\mathfrak{e}=\{e_1,\dots,e_d,e_{d+1},\dots,e_m\}\quad\text{and}\quad\mathfrak{f}=\{f_1,\dots,f_d,f_{d+1},\dots,f_m\}$$

are two such bases. Let $\mathbf{Q}$ be the change-of-basis matrix, which changes $\mathfrak{e}$ to $\mathfrak{f}$. The image of U under $\mathbf{Q}$ is obviously V, and thus $\mathbf{Q}\star\mathbf{x}\in V$. The existence of $\mathbf{P}$ is guaranteed by the following statement.

Claim. *If* $\mathbf{x}$ *and* $\mathbf{y}$ *have the same support and rank, then there exists a matrix* $\mathbf{P}\in\mathrm{GL}(n,q)$ *such that* $\mathbf{y}=\mathbf{x}\mathbf{P}$.

Proof. We prove the equality for φ_B, *i.e.,* there exists $\mathbf{P} \in \mathrm{GL}(n, q)$ such that $\varphi_B(\mathbf{y}) = \varphi_B(\mathbf{x})\mathbf{P}$. Without loss of generality, assume that $\mathrm{rank}(\mathbf{x}) = \mathrm{rank}(\mathbf{y}) = d$, and that the first d columns of both $\varphi_B(\mathbf{x})$ and $\varphi_B(\mathbf{y})$ are linearly independent over $\mathbb{F}_q$, all the rest $n-d$ columns of both matrices are the all-zero vectors. (These assumptions can be achieved by right-multiplying with matrices corresponding to the elementary column operations.)

Since the columns of $\varphi_B(\mathbf{x})$ and $\varphi_B(\mathbf{y})$ generate the same space, so by expressing each column of $\varphi_B(\mathbf{y})$ in terms of the columns of $\varphi_B(\mathbf{x})$, one get an invertible matrix $\mathfrak{P}$ of size $d \times d$. Now, it is clear that with $\mathbf{P} = \begin{pmatrix} \mathfrak{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-d} \end{pmatrix}$, one has $\varphi_B(\mathbf{x}) \cdot \mathbf{P} = \varphi_B(\mathbf{y})$. □

From the claim, (*iii*) is completely proven.

□

This operation can be regarded as the rank-equivalent permutation of the ordinary permutation, *i.e.,* permutation that permutes coordinates of vectors. Correspondingly, the rank-metric scheme is described as follows.

Common inputs: A matrix $\mathbf{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$, a syndrome $\mathbf{u} \in \mathbb{F}_{q^m}^{n-k}$, and a public hash function h.
Output: $\mathcal{P}$ proves in zero-knowledge that he knows a vector $\mathbf{s} \in \mathbb{F}_{q^m}^{n}$ of rank weight r such that $\mathbf{H}\mathbf{s}^T = \mathbf{u}^T$.

1. **Commitment.** The prover $\mathcal{P}$ randomly picks a vector $\mathbf{x}, \mathbf{P} \in \mathrm{GL}(n, q), \mathbf{Q} \in \mathrm{GL}(m, q)$, and computes

$$\begin{cases} c_1 = h\big(\mathbf{P}, \mathbf{Q}, \mathbf{H}\mathbf{x}^T\big), \\ c_2 = h\big(\mathbf{Q} \star \mathbf{x}\mathbf{P}\big), \\ c_3 = h\big(\mathbf{Q} \star (\mathbf{x} + \mathbf{s})\mathbf{P}\big). \end{cases}$$

   Then, $\mathcal{P}$ sends $cmt = (c_1, c_2, c_3)$ to the verifier $\mathcal{V}$.

2. **Challenge.** $\mathcal{V}$ randomly picks an element $b \in \{1, 2, 3\}$ and sends it to $\mathcal{P}$.

3. **Response.** $\mathcal{P}$ responses according to the value b.

   (a) If $b = 1$, $\mathcal{P}$ releases $\mathbf{Q} \star \mathbf{x}\mathbf{P}$ and $\mathbf{Q} \star \mathbf{s}\mathbf{P}$.

   (b) If $b = 2$, $\mathcal{P}$ releases $\mathbf{P}, \mathbf{Q}$ and $\mathbf{x} + \mathbf{s}$.

   (c) If $b = 3$, $\mathcal{P}$ releases $\mathbf{P}, \mathbf{Q}$ and $\mathbf{x}$.

4. **Verify.** $\mathcal{V}$ does the verification procedure as follows.

(a) If $b = 1$, $\mathcal{V}$ checks the values of c_2, c_3 and the condition $w(\mathbf{Q} \star \mathbf{sP}) = r$.

(b) If $b = 2$, $\mathcal{V}$ checks the values of c_1 and c_3.

(c) If $b = 3$, $\mathcal{V}$ checks the values of c_1 and c_2.

If all checks pass, then $\mathcal{V}$ outputs 1; otherwise, it outputs 0.

Chapter 3

Chameleon Hash Signatures

One method to design digital signatures is the use of hash-and-sign paradigm. The use of hash functions in a scheme means that the security of the designed scheme is being considered in the random oracle model. In post-quantum code-based cryptography, security against quantum adversary in the (quantum) random oracle model is still unknown. Thus, it would be more desirable if the scheme does not make use of hash functions, *i.e.,* truly random functions.

Motivated by this task, that is, constructing a code-based signature scheme without the use of hash functions, we designed a type of functions from standard code-based assumptions, which has some features similar to those of a hash function, and therefrom, succeeded in deriving a code-based signature scheme. This kind of functions is called *chameleon hash function,* whose notion was first introduced by Krawczyk and Rabin [KR00].

This work is a joint work with Olivier Blazy, Philippe Gaborit, Ayoub Otmani, and Jean-Pierre Tillich, and was presented in the International Workshop on Coding and Cryptography 2018 (WCC 2018).

3.1 Introduction

In 1997, Kabatiansky, Krouk, and Smeets proposed in [KKS97] a signature scheme based on the difficulty of decoding an $[N, K]$ binary random code. The public key of the scheme consists in a parity-check matrix of this code together with k syndromes of errors whose supports are all included in a small support of size n. The corresponding errors form the secret key of the scheme. They allow to sign a binary message of length k by taking the corresponding linear combinations of these errors. This linear combination is typically an error of rather small weight (since it has weight $\approx \frac{n}{2}$) whose syndrome can be computed by a verifier from the k public syndromes. Later on, several variants of this scheme were proposed

[KKS05, BMS11].

However, in 2011, Otmani and Tillich [OT11] devised attacks against these schemes. They showed that if $\frac{k}{n}$ is not significantly smaller than $\frac{K}{N}$, then there is an efficient attack on these schemes. This did not undermine the security of the whole scheme since the attack is still exponential in nature but just showed that the parameters of the scheme have to be chosen carefully.

On the other direction of research, starting with the results of Krawczyk and Rabin [KR00], the work of Bellare and Shoup [BS07], and Blazy *et al.* [BKKP15], another method of constructing and a new notion of security of signature scheme are proposed, *i.e., chameleon signature* and *two-tier* security. Briefly speaking, a chameleon signature scheme consists of two ingredients: a chameleon hash function, and a regular signature scheme. In this type of signature scheme, the power of the recipient (*i.e.,* possessing the trapdoor of the chameleon hash function) gives the scheme extra properties such as *non-transferability* and *non-repudiation*. In [BKKP15], this power is given to the signer to strengthen the security, that is, their scheme achieves two-tier security in the standard model.

In this work, we combine the two directions to (i) construct a chameleon hash function from the KKS assumption, and (ii) devise a code-based chameleon-hash signature scheme using this function and also derive a corresponding binary tree-based scheme by using the methods of [BKKP15]. This gives the first code-based signature scheme with a security proof in the *standard model*. It is also worthwhile to recall that obtaining an efficient and provably secure scheme in the much weaker random oracle model is already quite a formidable challenge as illustrated by the fact that all the recent code-based signature schemes to the NIST competition for standardizing post-quantum public key cryptography were broken. The signature we propose here is also a post-quantum candidate and the only other candidates for being secure against a quantum computer in the standard model are lattice based signature schemes using bonsai trees and variations of this approach.

The rest of this work is organized as follows. In Section 3.2, we recall basic facts on signature schemes as well as some hard problems in coding theory; in Section 3.3, we construct a chameleon hash function whose security is based on the KKS scheme and other hard problems from coding theory; in Section 3.4, we derive a signature scheme using the constructed chameleon hash function using the techniques in [BKKP15]; and in the two last sections, we give some concrete parameters for the scheme and draw some conclusions.

3.2 Preliminaries

3.2.1 Notation

Throughout the work, vectors are written in row form and denoted by bold lowcase letters whereas matrices are denoted by bold capital letters. For a given vector $\mathbf{v}$ and a subset J of indices, we let $\mathbf{v}_J = (v_j)_{j\in J}$, the HAMMING weight of $\mathbf{v}$ is denoted by $\|\mathbf{v}\|$; its transpose is denoted by $\mathbf{v}^T$; a similar notation is used for the transpose of a matrix. By writing $x \leftarrow X$, we mean that x is drawn according to the distribution X, if X is a distribution; or drawn uniformly at random from X when X is a set, or the output of the algorithm X, if X is an algorithm. For two probability distributions A and B, $A \equiv B$ means that the two distributions are identical, and $A \stackrel{c}{\equiv} B$ means that A and B are computationally indistinguishable.

For a variable $x \in (0,1)$, the binary entropy function of x is denoted by $h_2(x)$. Here, we recall that $h_2(x) = -x\log_2 x - (1-x)\log_2(1-x)$.

3.2.2 Signatures

We recall here the definition of a digital signature scheme.

Definition 3.1 (Signature scheme)**.** *A digital signature scheme* Sig *with message space* $\mathcal{M}$ *is a triple of probabilistic polynomial-time algorithms,* $\mathsf{Sig} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$*, that satisfies:*

- *On input* 1^λ*, algorithm* Gen *outputs a signing key* sk *and a verification key* pk*.*
- *On input a signing key* sk *and a message* $m \in \mathcal{M}$*, algorithm* Sign *outputs a signature* σ*.*
- *On input consisting of a public key and a message-signature pair* (m, σ)*, algorithm* Verify *outputs* 1 (accept) *or* 0 (reject)*.*

Sig *is correct if for any positive integer* λ*, all* $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$*, all* $m \in \mathcal{M}$*, and all* $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$*, it holds that* $\mathsf{Verify}(\mathsf{pk}, m, \sigma) = 1$*.*

For the security of a signature scheme, we consider the notion of existentially unforgeability.

Definition 3.2. *A signature scheme, denoted by* Sig*, is* (t, ε, q)*-existential unforgeable under non-adaptive chosen-message attacks* (EUF-NCMA) *if*

$$\Pr\left[\mathsf{Exp}^{\mathsf{EUF\text{-}NCMA}}_{\mathsf{Sig},\mathcal{F},q}(\lambda) = 1\right] \leq \varepsilon$$

holds for any probabilistic polynomial-time adversary $\mathcal{F}$ with running time t and q signature queries, where $\mathsf{Exp}^{\mathsf{EUF\text{-}NCMA}}_{\mathsf{Sig},\mathcal{F},q}(\lambda)$ is defined as follows. (We also give the definition of Existential unforgeability under chosen-message attacks.)

Experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}NCMA}}_{\mathsf{Sig},\mathcal{F},q}(\lambda)$

$\mathcal{Q} := (m_1, \ldots, m_q) \leftarrow \mathcal{F}(\lambda)$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
for $i = 1$ **to** q **do**
  $\sigma_i \leftarrow \mathsf{Sign}(\mathsf{sk}, m_i)$
end for
$(m^*, \sigma^*) \leftarrow \mathcal{F}(\mathsf{pk}, \sigma_1, \ldots, \sigma_q)$
if $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*) = 1$ **and** $m^* \notin \mathcal{Q}$ **then**
  return 1
else
  return 0
end if

Experiment $\mathsf{Exp}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{Sig},\mathcal{F},q}(\lambda)$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$
$(m^*, \sigma^*) \leftarrow \mathcal{F}^{\mathsf{Sign}(\mathsf{sk},\cdot)}(\mathsf{pk})$ with $\mathcal{Q} := \{m_1, \ldots, m_q\}$, where m_i is the i-th message queried
if $\mathsf{Verify}(\mathsf{pk}, m^*, \sigma^*) = 1$ **and** $m^* \notin \mathcal{Q}$ **then**
  return 1
else
  return 0
end if

The notion of security which is stronger than the notion EUF is called strong unforgeability, SUF. In the SUF experiment, the adversary is allowed to forge a new signature on a message which has been already challenged. In order to do that, we set $\mathcal{Q} = \{(m_1, \sigma_1), \ldots, (m_q, \sigma_q)\}$. Now, a valid forgery is a pair $(m^*, \sigma^*) \notin \mathcal{Q}$. This notion is applied for both adaptive and non-adaptive security, and are refered to as SUF-CMA and SUF-NCMA, respectively.

3.2.3 Two-Tier Signatures

We recall the notion of two-tier signature schemes due to Bellare and Shoup [BS07]. In a two-tier signature scheme, the key generation algorithm is split into two algorithms: the primary (PriGen) and the secondary (SecGen) key generation algorithms. The primary key is static and used for all signatures. The secondary key is ephemeral and used for only one signature. The following definitions are from [BKKP15], which is a generalization of two-tier signature.

Definition 3.3 (d-time two-tier signature scheme)**.** *A* two-tier *signature scheme,* TTSig*, is a quadruple of probabilistic polynomial-time algorithms,* TTSig = (PriGen, SecGen, TTSign, TTVerify)*, satisfying that:*

- *On input* $1^\lambda, d$*,* PriGen *outputs a primary signing key* psk *and a primary verification key* ppk*.*

- *On input* ppk *and* psk, SecGen *outputs a fresh verification and signing key pair* $(\mathsf{spk}, \mathsf{ssk})$.
- *On input* $\mathsf{psk}, \mathsf{ssk}$ *and a message* m, *algorithm* TTSign *outputs a signature* σ. *We denote the stateful variant by* $\mathsf{TTSign}(\mathsf{psk}, \mathsf{ssk}, m; j)$, *where* $1 \leq j \leq d$ *is the state.*
- *On input* $\mathsf{ppk}, \mathsf{spk}$, *a message* m *and a signature* σ, *algorithm* $\mathsf{TTVerify}$ *deterministically outputs* 1 (accept) *or* 0 (reject). *We denote the stateful variant by* $\mathsf{TTVerify}(\mathsf{ppk}, \mathsf{spk}, m, \sigma; j)$.

Security of two-tier signature scheme is stated in the following definition.

Definition 3.4 (Security of two-tier signature scheme)**.** *A two-tier signature scheme* TTSig *is* (t, q, d, ε)*-existential unforgeable under non-adaptive chosen-message attacks* (TT-EUF-NCMA) *if*

$$\Pr\left[\mathsf{Exp}^{\text{TT-EUF-NCMA}}_{\mathsf{TTSign},\mathcal{F},q}(\lambda, d) = 1\right] \leq \varepsilon$$

holds for any probabilistic polynomial-time adversary $\mathcal{F}$ *with running time* t, *where* $\mathsf{Exp}^{\text{TT-EUF-NCMA}}_{\mathsf{TTSign},\mathcal{F},q}(\lambda, d)$ *is defined in Table 3.1. Existential unforgeability under adaptive chosen-message attacks* (TT-EUF-CMA) *is defined similarly.*

Experiment $\mathsf{Exp}^{\text{TT-EUF-NCMA}}_{\mathsf{TTSign},\mathcal{F},q}(\lambda, d)$: $(\mathsf{ppk}, \mathsf{psk}) \leftarrow \mathsf{PriGen}(1^\lambda, d)$; $(m^*, \sigma^*, i^*) \leftarrow \mathcal{F}^{\mathtt{NTTSign}(\cdot)}(\mathsf{ppk})$; If $\mathsf{TTVerify}(\mathsf{ppk}, \mathsf{spk}_{i^*}, m^*, \sigma^*) = 1$ and $m^* \notin \mathcal{Q}_{i^*}$, then return 1, else return 0.	**Experiment** $\mathsf{Exp}^{\text{TT-EUF-CMA}}_{\mathsf{TTSign},\mathcal{F},q}(\lambda, d)$: $(\mathsf{ppk}, \mathsf{psk}) \leftarrow \mathsf{PriGen}(1^\lambda, d)$; $(m^*, \sigma^*, i^*) \leftarrow \mathcal{F}^{\mathtt{OSKey}(), \mathtt{TTSign}(\cdot,\cdot)}(\mathsf{ppk})$; If $\mathsf{TTVerify}(\mathsf{ppk}, \mathsf{spk}_{i^*}, m^*, \sigma^*) = 1$ and $m^* \notin \mathcal{Q}_{i^*}$, then return 1, else return 0.
Oracle $\mathtt{NTTSign}(m_1, \ldots, m_d)$: $i = i + 1$ and $(\mathsf{spk}_i, \mathsf{ssk}_i) \leftarrow \mathsf{SecGen}(\mathsf{ppk}, \mathsf{psk})$; $\sigma_j \leftarrow \mathsf{TTSign}(\mathsf{psk}, \mathsf{ssk}_i, m_j)$ for $j = 1, \ldots, d$; Store $(m_1, \ldots, m_d)$ in the list $\mathcal{Q}_i$; Return $(\mathsf{spk}_i, \sigma_1, \ldots, \sigma_d)$.	**Oracle** $\mathtt{OSKey}()$: $i = i + 1$ and $j_i = 0$; $(\mathsf{spk}_i, \mathsf{ssk}_i) \leftarrow \mathsf{SecGen}(\mathsf{ppk}, \mathsf{psk})$; Return spk_i. **Oracle** $\mathtt{TTSign}(i', m)$: $j_{i'} = j_{i'} + 1$; $m_{j_{i'}} := m$ If $j_{i'} > d$ or $(\mathsf{spk}_{i'}, \mathsf{ssk}_{i'})$ is undefined then return $\perp$; $\sigma \leftarrow \mathsf{TTSign}(\mathsf{psk}, \mathsf{ssk}_{i'}, m_{j_{i'}})$ and store $m_{j_{i'}}$ in $\mathcal{Q}_{i'}$; Return σ.

Table 3.1: TT-EUF-NCMA and TT-EUF-CMA experiments for two-tier signature scheme.

Here, $\mathcal{F}^{\mathcal{O}}$ means that $\mathcal{F}$ is given access to oracle $\mathcal{O}$. The strong unfogeability security of two-tier signatures, *i.e.,* TT-SUF-CMA and TT-SUF-NCMA, are defined in the same ways as in the standard signatures.

3.2.4 Chameleon Hash Functions

The notion of chameleon hash function was introduced by Krawczyk and Rabin [KR00]. Here, we briefly recall the definition and some of its properties.

Definition 3.5. *A* chameleon hash *function is defined as* $\mathsf{CHF} = (\mathsf{CHGen}, \mathsf{CHash}, \mathsf{Coll})$*, where:*

- $\mathsf{CHGen}(1^\lambda)$ *outputs a hash key* chk *and the corresponding trapdoor* td*.*
- $\mathsf{CHash}(\mathsf{chk}, m, r)$ *outputs the hash value* h *on a message* m *and a randomness* r*.*
- $\mathsf{Coll}(\mathsf{td}, (m, r), \widehat{m})$ *outputs a randomness* $\widehat{r}$ *such that*

$$\mathsf{CHash}(\mathsf{chk}, m, r) = \mathsf{CHash}(\mathsf{chk}, \widehat{m}, \widehat{r}).$$

Security of chameleon hash function is stated as follows.

Definition 3.6. *A chameleon hash function* CHF *is said to be* (t, ε)*-collision resistant if for an adversary* $\mathcal{A}$ *running in time at most* t*, it holds that*

$$\Pr_{\substack{(\mathsf{chk},\mathsf{td})\leftarrow\mathsf{CHGen}(1^\lambda) \\ ((m_1,r_1)\neq(m_2,r_2))\leftarrow\mathcal{A}(\mathsf{chk})}} \left[\mathsf{CHash}(\mathsf{chk}, m_1, r_1) = \mathsf{CHash}(\mathsf{chk}, m_2, r_2)\right] \leq \varepsilon.$$

A chameleon hash function $\mathsf{CHash}(\mathsf{chk}, \cdot, \cdot)$ with hash key chk and corresponding trapdoor td has to meet the following properties:

1. **Collision resistance:** There is no efficient algorithm that can find two pairs (m_1, r_1) and (m_2, r_2) with $m_1 \neq m_2$ such that

$$\mathsf{CHash}(\mathsf{chk}, m_1, r_1) = \mathsf{CHash}(\mathsf{chk}, m_2, r_2).$$

2. **Trapdoor collision:** Given td, there exists an efficient algorithm that on any pair (m_1, r_1) and a message $m_2 \neq m_1$ finds a value r_2 such that

$$\mathsf{CHash}(\mathsf{chk}, m_1, r_1) = \mathsf{CHash}(\mathsf{chk}, m_2, r_2).$$

3. **Uniformity:** All messages m induce the same probability distribution on $\mathsf{CHash}(\mathsf{chk}, \cdot, \cdot)$ for r chosen randomly. This statement can be relaxed to require that the distributions induced by different messages are *computationally indistinguishable*. That is, for two given messages m_1, m_2, we require that

$$\{\mathsf{CHash}(\mathsf{chk}, m_1, r_1) \mid r_1 \leftarrow \mathcal{D}_1\} \stackrel{c}{\equiv} \{\mathsf{CHash}(\mathsf{chk}, m_2, r_2) \mid r_2 \leftarrow \mathcal{D}_2\},$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are two appropriate distributions.

3.2.5 Difficult Problems

Besides Problem 1 and Problem 3, which are stated in Section 2.4.1, we also consider the case when the weight of errors varies in an *acceptable interval, i.e.,* an interval in which the decoding is still hard.

Definition 3.7 (Extended Syndrome Decoding Distribution)**.** *For positive integers n, k, a, b, $a \leq b$, the extended syndrome decoding distribution,* $\mathsf{extSD}(n, k, a, b)$*, chooses* $\mathbf{H} \leftarrow \mathbb{F}_q^{(n-k)\times n}$ *and* $\mathbf{x} \leftarrow \mathbb{F}_q^n$ *such that* $a \leq \|\mathbf{x}\| \leq b$*, and outputs* $(\mathbf{H}, \sigma(\mathbf{x}) = \mathbf{H} \cdot \mathbf{x}^T)$*.*

Definition 3.8 (Extended Decisional Syndrome Decoding Problem)**.** *On an input* $(\mathbf{H}, \mathbf{y}^T) \leftarrow \mathbb{F}_q^{(n-k)\times n} \times \mathbb{F}_q^{n-k}$*, the decision* extSD *problem,* $\mathsf{extDSD}(n, k, a, b)$*, asks to decide with non-negligible advantage whether* $(\mathbf{H}, \mathbf{y}^T)$ *came from the* $\mathsf{extSD}(n, k, a, b)$ *distribution or the uniform distribution over* $\mathbb{F}_q^{(n-k)\times n} \times \mathbb{F}_q^{n-k}$*.*

Here, we make the assumption that the $\mathsf{extDSD}(n, k, a, b)$ problem is hard. In [KKS97], besides providing that the weight of the error lies in the interval $[t_1, t_2]$, the authors also reveal a matrix $\mathbf{F}$, which is closely related to the matrix $\mathbf{H}$. The problem above just provides the information on the weight of the error and nothing more, and in fact, it can be seen as a corollary of Conjecture 3 in [Ale03].

3.3 The Transformation

3.3.1 The KKS Scheme

The KKS scheme uses two codes: a linear code defined by an $(N-K) \times N$ parity-check matrix $\mathbf{H}$ over $\mathbb{F}_q$ (in most cases $q = 2$); a linear code $\mathcal{C}_{\text{hid}}$ over $\mathbb{F}_q$ of length $n \leq N$ and of dimension k which is defined by a $k \times n$ generator matrix $\mathbf{G}$. The code $\mathcal{C}_{\text{hid}}$ has the property that there exist two positive integers $t_1 \leq t_2$ such that with high probability, $t_1 \leq \|\mathbf{c}\| \leq t_2$ for any non-zero codeword $\mathbf{c} \in \mathcal{C}_{\text{hid}}$. The description of the scheme is as follows.

1. $\mathsf{Gen}(1^\lambda)$: The signer
   - chooses parameters N, K, n, k, t_1, and t_2 with respect to the security parameter λ;
   - draws a random $(N - K) \times N$ matrix $\mathbf{H}$; chooses an n-subset $J \subset \{1, \ldots, N\}$;
   - chooses a random $k \times n$ generator matrix $\mathbf{G}$ that defines a code $\mathcal{C}_{\text{hid}}$ such that with high probability, $t_1 \leq \|\mathbf{c}\| \leq t_2$ for any non-zero codeword $\mathbf{c} \in \mathcal{C}_{\text{hid}}$;

- defines $\mathbf{F} \stackrel{\text{def}}{=} \mathbf{H}_J\mathbf{G}^T$, where $\mathbf{H}_J$ is the restriction of $\mathbf{H}$ to the columns in J;
- publishes $\mathbf{H}$ and $\mathbf{F}$ as the public key pk, and keeps J and $\mathbf{G}$ as the secret key sk.

2. $\mathsf{Sign}(\mathsf{sk}, \mathbf{x})$:
   - On input a message $\mathbf{x} \in \mathbb{F}_q^k$, the signer computes $\mathbf{v} = \mathbf{x} \cdot \mathbf{G}$.
   - Next, the signer defines the signature $\mathbf{s} = (s_1, \ldots, s_N)$ as $\mathbf{s}_J = \mathbf{v}$, and $s_i = 0$ for $i \notin J$.

3 $\mathsf{Verify}(\mathsf{pk}, (\mathbf{x}, \mathbf{s}))$: On input a pair $(\mathbf{x}, \mathbf{s}) \in \mathbb{F}_q^k \times \mathbb{F}_q^N$, the verifier checks that $t_1 \leq \|\mathbf{s}\| \leq t_2$, and $\mathbf{H} \cdot \mathbf{s}^T = \mathbf{F} \cdot \mathbf{x}^T$.

As noticed in [KKS97], the code $\mathcal{C}_{\text{hid}}$ can be chosen as a random code. A signature corresponds to a random codeword of $\mathcal{C}_{\text{hid}}$ and it is readily seen that its weight lies in an interval $[t_1, t_2]$ with high probability. This probability is estimated as in the following propositions.

Proposition 3.1. *Let $\mathcal{C} = [n, k]$ be a random binary code and $0 < d \leq \frac{n}{2}$ a positive integer. The probability that the minimum distance $d_{\mathcal{C}} \geq d$ is at least $1 - 2^{-(n-k)+n\cdot h_2\left(\frac{d-1}{n}\right)}$.*

Proof. In this proof, an $[n, k]$ binary code is identical with a subspace of $\mathbb{F}_2^n$ of dimension k. Define

$$B_r(\mathbf{x}) = \left\{\mathbf{v} \in \mathbb{F}_2^n \mid \|\mathbf{x} - \mathbf{v}\| \leq r\right\}.$$

For a given vector $\mathbf{0} \neq \mathbf{x} \in \mathbb{F}_2^n$, we have

Claim. *The number of subspaces of dimension k which contain $\mathbf{x}$ is $\frac{(2^n-2)\cdots(2^n-2^{k-1})}{(2^k-2)\cdots(2^k-2^{k-1})}$.*

Recall that the number of subspaces of $\mathbb{F}_2^n$ of dimension k is

$$\frac{(2^n-1)(2^n-2)\cdots(2^n-2^{k-1})}{(2^k-1)(2^k-2)\cdots(2^k-2^{k-1})}.$$

Putting these together, we see that the probability that a randomly chosen subspace V of dimension k contains $\mathbf{x}$ is $\frac{2^k-1}{2^n-1}$. Thus, the probability that a randomly chosen subspace V of dimension k does not contain $\mathbf{x}$ is

$$\begin{aligned}\Pr[\mathbf{x} \notin V] &= 1 - \frac{2^k-1}{2^n-1} \\ &\geq 1 - \frac{1}{2^{n-k}}.\end{aligned}$$

From this estimation, it follows that for a given subset T of $\mathbb{F}_2^n$, which does not contain $\mathbf{0}$, the probability that a random subspace of dimension k does not intersect T satisfies

$$\Pr[V \cap T = \varnothing] \geq \left(1 - \frac{1}{2^{n-k}}\right)^{|T|}. \tag{3.1}$$

Therefore, the probability that a randomly chosen subspace V of dim $= k$ does not have common (non-zero) elements with $B_{d-1}(\mathbf{0})$ satisfies

$$\begin{aligned}\Pr[V \cap B_{d-1}(\mathbf{0}) = \{\mathbf{0}\}] &\geq \left(1 - \frac{1}{2^{n-k}}\right)^{|B_{d-1}(\mathbf{0})|-1} \\ &\geq 1 - \frac{|B_{d-1}(\mathbf{0})| - 1}{2^{n-k}} \\ &\geq 1 - \frac{|B_{d-1}(\mathbf{0})|}{2^{n-k}}.\end{aligned}$$

The statement of the theorem is implied from the following lemma.

Lemma 3. *Let $t \leq \frac{n}{2}$ be a positive integer. Then,*

$$|B_t(\mathbf{0})| \leq 2^{n \cdot h_2\left(\frac{t}{n}\right)}.$$

Proof. Note that one has

$$|B_t(\mathbf{0})| = \binom{n}{0} + \cdots + \binom{n}{t},$$

and $h_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ for $0 < x < 1$. Thus, the inequality is equivalent to

$$\binom{n}{0} + \cdots + \binom{n}{t} \leq \frac{n^n}{t^t \cdot (n-t)^{n-t}}.$$

Since $t \leq \frac{n}{2}$, so for all $0 \leq i \leq t$, we have

$$t^t \cdot (n-t)^{n-t} \leq t^i \cdot (n-t)^{n-i}.$$

Hence,

$$t^t \cdot (n-t)^{n-t} \cdot \left\{\binom{n}{0} + \cdots + \binom{n}{t}\right\} \leq \sum_{i=0}^{t} \binom{n}{i} t^i \cdot (n-t)^{n-i}.$$

The sum on the right-hand side is obviously less than n^n. □

□

The above proposition first appeared in [KKS97] without proof. Here, it is supplied with a proof. The next proposition is a direct corollary.

Proposition 3.2. *Let $\mathcal{C} = [n, k]$ be a random binary code in systematic form and $0 < t_1 \leq \frac{n}{2} < t_2 \leq n$. Then the probability that all codewords of $\mathcal{C}$ lie in the interval $\left[t_1, t_2\right]$ is at least*

$$1 - 2^{-(n-k)+n\cdot h_2\left(\frac{t_1-1}{n}\right)} - 2^{-(n-k)+n\cdot h_2\left(\frac{n-t_2-1}{n}\right)}.$$

Proof. In the proof, V is always understood to be a randomly chosen subspace of $\mathbb{F}_2^n$ of dimension k. For $\frac{n}{2} < t \leq n$, let $\mathbf{1} = (1, 1, \ldots, 1)$ and

$$B_{n-t}(\mathbf{1}) = \{\mathbf{x} \in \mathbb{F}_2^n \mid d(\mathbf{x}, \mathbf{1}) \leq n - t\}.$$

Obviously, we have

$$B_{n-t}(\mathbf{1}) = \{\mathbf{x} \in \mathbb{F}_2^n \mid w(\mathbf{x}) \geq t\}.$$

By Lemma 3, we have

$$|B_{n-t}(\mathbf{1})| \leq 2^{n\cdot h_2\left(\frac{n-t}{n}\right)}.$$

Note that the event $V \cap B_{n-t}(\mathbf{1}) = \varnothing$ is equivalent to $V \subseteq B_{t-1}(\mathbf{0})$. By letting $t = t_2 + 1$ and

$$T = B_{t_1}(\mathbf{0}) \cup B_{n-t_2-1}(\mathbf{1}) \setminus \{\mathbf{0}\},$$

we see that the event $V \subseteq (B_{t_2}(\mathbf{0}) \setminus B_{t_1}(\mathbf{0})) \cup \{\mathbf{0}\}$ is equivalent to $V \cap T = \varnothing$. By applying Equation 3.1 and observe that

$$|T| \leq 2^{n\cdot h_2\left(\frac{t_1-1}{n}\right)} + 2^{n\cdot h_2\left(\frac{n-t_2-1}{n}\right)} - 1,$$

one gets

$$\begin{aligned}
\Pr[V \cap T = \varnothing] &\geq \left(1 - \frac{1}{2^{n-k}}\right)^{2^{n\cdot h_2\left(\frac{t_1-1}{n}\right)} + 2^{n\cdot h_2\left(\frac{n-t_2-1}{n}\right)} - 1} \\
&\geq \left(1 - \frac{1}{2^{n-k}}\right)^{2^{n\cdot h_2\left(\frac{t_1-1}{n}\right)}} \cdot \left(1 - \frac{1}{2^{n-k}}\right)^{2^{n\cdot h_2\left(\frac{n-t_2-1}{n}\right)}} \\
&\geq \left(1 - 2^{-(n-k)+n\cdot h_2\left(\frac{t_1-1}{n}\right)}\right)\left(1 - 2^{-(n-k)+n\cdot h_2\left(\frac{n-t_2-1}{n}\right)}\right) \\
&\geq 1 - 2^{-(n-k)+n\cdot h_2\left(\frac{t_1-1}{n}\right)} - 2^{-(n-k)+n\cdot h_2\left(\frac{n-t_2-1}{n}\right)}.
\end{aligned}$$

Therefore,

$$\Pr[V \subseteq (B_{t_2}(\mathbf{0}) \setminus B_{t_1}(\mathbf{0})) \cup \{\mathbf{0}\}] \geq 1 - 2^{-(n-k)+n\cdot h_2\left(\frac{t_1-1}{n}\right)} - 2^{-(n-k)+n\cdot h_2\left(\frac{n-t_2-1}{n}\right)}.$$

The proposition is proven.

□

As mentioned in the introduction, the original KKS schemes with its proposed parameters (and some other variants) were efficiently attacked by Otmani and Tillich in [OT11]. However, as already pointed out in [OT11], this attack is of exponential nature and can be avoided if the parameters are chosen carefully. We refer to Section 3.5 for such a selection. Therefore, we make use of the following assumption.

Assumption 1 (KKS assumption)**.** *There is some region of parameters such that the above scheme is one-time* EUF-CMA.

3.3.2 A Chameleon Hash Function

In this section, we construct a chameleon hash function in the relaxed sense using the KKS assumption. First, define two types of sets as

$$\mathcal{S}_d \stackrel{\text{def}}{=} \{\, \mathbf{s} \in \mathbb{F}_q^N \mid \|\mathbf{s}\| = d \,\},$$
$$\mathcal{S}_{[a,b]} \stackrel{\text{def}}{=} \{\, \mathbf{s} \in \mathbb{F}_q^N \mid a \leq \|\mathbf{s}\| \leq b \,\}.$$

Now, we consider the function $f(\mathsf{ppk}, \cdot, \cdot)\colon \mathbb{F}_q^k \times \mathcal{S}_t \longrightarrow \mathbb{F}_q^{N-K}$ defined as

$$f(\mathsf{ppk}, \mathbf{x}, \mathbf{s}) \stackrel{\text{def}}{=} \mathbf{F} \cdot \mathbf{x}^T + \mathbf{H} \cdot \mathbf{s}^T,$$

where $\mathsf{ppk} = (\mathbf{F}, \mathbf{H})$ comes from a KKS signature scheme, $\mathbf{x} \in \mathbb{F}_q^k$ random, and $\mathbf{s} \in \mathcal{S}_t$, where t is a positive integer which is defined later. On input a pair $(\mathbf{x}_1, \mathbf{s}_1) \in \mathbb{F}_q^k \times \mathcal{S}_t$ and a message $\mathbf{x}_2 \neq \mathbf{x}_1 \in \mathbb{F}_q^k$, the one who possesses $J, \mathbf{G}$ (*i.e.,* the *trapdoor*) can find an $\mathbf{s}_2 \in \mathcal{S}_{[t-t_2, t+t_2]}$, with assumption that $t \geq t_2$, such that $f(\mathsf{ppk}, \mathbf{x}_1, \mathbf{s}_1) = f(\mathsf{ppk}, \mathbf{x}_2, \mathbf{s}_2)$ as follows:

1. Compute $\mathbf{v}^T = \mathbf{x}_1^T - \mathbf{x}_2^T \in \mathbb{F}_q^k$.
2. Solve the equation $\mathbf{F} \cdot \mathbf{v}^T = \mathbf{H} \cdot \mathbf{s}^T$ for $\mathbf{s} \in \mathcal{S}_{[t_1, t_2]}$ (using the signing process of the KKS scheme and set $\mathbf{s}$ to be the signature corresponding to $\mathbf{v}$).
3. Define $\mathbf{s}_2 = \mathbf{s} + \mathbf{s}_1 \in \mathbb{F}_q^N$. It can be seen that $\|\mathbf{s}_2\| \leq \|\mathbf{s}\| + \|\mathbf{s}_1\| \leq t_2 + t$, and $\|\mathbf{s}_2\| \geq t - t_2$.

It is clear that

$$\begin{aligned} \mathbf{F} \cdot \mathbf{x}_2^T + \mathbf{H} \cdot \mathbf{s}_2^T &= \mathbf{F} \cdot \mathbf{x}_2^T + \mathbf{H} \cdot (\mathbf{s} + \mathbf{s}_1)^T \\ &= \mathbf{F} \cdot (-\mathbf{v}^T + \mathbf{x}_1^T) + \mathbf{H} \cdot \mathbf{s}^T + \mathbf{H} \cdot \mathbf{s}_1^T \\ &= \mathbf{F} \cdot \mathbf{x}_1^T + \mathbf{H} \cdot \mathbf{s}_1^T. \end{aligned}$$

A collision is the two pairs $(\mathbf{x}_1, \mathbf{s}_1) \neq (\mathbf{x}_2, \mathbf{s}_2)$ with $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{F}_q^k$, $\mathbf{s}_1 \in \mathcal{S}_t$, and $\mathbf{s}_2 \in \mathcal{S}_{[t-t_2,t+t_2]}$. For the uniformity, given $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{F}_q^k$, the uniform property requires that when $\mathbf{s}_1 \leftarrow \mathcal{S}_t$ and $\mathbf{s}_2 \leftarrow \mathcal{S}_{[t-t_2,t+t_2]}$, the induced probability distributions are computationally indistinguishable, *i.e.,*

$$\left\{ f(\mathsf{ppk}, \mathbf{x}_1, \mathbf{s}_1) \mid \mathbf{s}_1 \leftarrow \mathcal{S}_t \right\} \stackrel{c}{\equiv} \left\{ f(\mathsf{ppk}, \mathbf{x}_2, \mathbf{s}_2) \mid \mathbf{s}_2 \leftarrow \mathcal{S}_{[t-t_2,t+t_2]} \right\}.$$

We claim that the function f is a chameleon hash function exactly in this sense.

Proposition 3.3. *Let $N, K, t, t_1, t_2, \lambda$ be positive integers such that $t_1 \leq t_2 \leq t$ and*

$$N - K - N \cdot h_2\left(\frac{4t + 2t_2}{N}\right) > \lambda.$$

Assume that the $\mathsf{DSD}(N, K, t)$ and the $\mathsf{extDSD}(N, K, t - t_2, t + t_2)$ problems are hard, and the KKS signature scheme for $t_1, t_2, \mathbf{F}, \mathbf{H}$ is one-time secure with additional property that each message has unique signature, then f is a chameleon hash function.

Proof. We need to show that the function f defined as above satisfies the three properties of a chameleon hash function.

Collision resistance: This property is guaranteed by the KKS assumption. If an adversary can break the KKS assumption, then he can find a collision as above.

Now, assume that there exists an adversary $\mathcal{A}$ who, given $(\mathbf{F}, \mathbf{H})$, could find two pairs $(\mathbf{x}_1, \mathbf{s}_1) \in \mathbb{F}_q^k \times \mathcal{S}_t$ and $(\mathbf{x}_2, \mathbf{s}_2) \in \mathbb{F}_q^k \times \mathcal{S}_{[t-t_2,t+t_2]}$, $\mathbf{x}_1 \neq \mathbf{x}_2$ with probability ε such that

$$\mathbf{F} \cdot \mathbf{x}_1^T + \mathbf{H} \cdot \mathbf{s}_1^T = \mathbf{F} \cdot \mathbf{x}_2^T + \mathbf{H} \cdot \mathbf{s}_2^T. \tag{3.2}$$

We construct an algorithm $\mathcal{F}$ that breaks the KKS signature scheme with probability ε. After receiving the pair of matrices $(\mathbf{F}, \mathbf{H})$ from the KKS scheme, algorithm $\mathcal{F}$ proceeds as follows:

1. Send $(\mathbf{F}, \mathbf{H})$ to $\mathcal{A}$.

2. On receiving two pairs $(\mathbf{x}_1, \mathbf{s}_1), (\mathbf{x}_2, \mathbf{s}_2)$ from $\mathcal{A}$, compute $\sigma = \mathbf{s}_1 - \mathbf{s}_2$.

3. Output $(\mathbf{x}_2 - \mathbf{x}_1, \sigma)$ as a pair of forged message-signature for the KKS scheme.

We show that $(\mathbf{x}_2 - \mathbf{x}_1, \sigma)$ is a legitimate pair of message-signature for the KKS scheme, *i.e.,* $\sigma \in \mathcal{S}_{[t_1,t_2]}$. Let $\mathbf{v} = \mathbf{x}_2 - \mathbf{x}_1$ be regarded as known and consider the following equation in the unknown $\mathbf{s}$

$$\mathbf{F} \cdot \mathbf{v}^T = \mathbf{H} \cdot \mathbf{s}^T \quad \text{for} \quad 0 \leq \|\mathbf{s}\| \leq 2t + t_2. \tag{3.3}$$

Since $\mathbf{s}_1 \in \mathcal{S}_t$ and $\mathbf{s}_2 \in \mathcal{S}_{[t-t_2,t+t_2]}$ so

$$\begin{aligned}\|\mathbf{s}_1 - \mathbf{s}_2\| &\geq 0,\\ \|\mathbf{s}_1 - \mathbf{s}_2\| &\leq t + (t + t_2) = 2t + t_2.\end{aligned}$$

Thus σ is a solution of Equation 3.3. From Proposition 3.1, the code defined by $\mathbf{H}$ has minimum distance greater than $2(2t + t_2)$ with probability at least

$$1 - 2^{-N+K+N\cdot h_2\left(\frac{4t+2t_2}{N}\right)} \geq 1 - 2^{-\lambda}.$$

As a consequence, Equation 3.3 has unique solution with probability at least $1 - 2^{-\lambda}$. On the other hand, we have already known that this equation has a solution $\mathbf{s} \in \mathcal{S}_{[t_1,t_2]}$, which can be found by using the trapdoor on the pair $(\mathbf{x}_1, \mathbf{s}_1)$ and $\mathbf{x}_2$. From these arguments, we deduce that

$$\sigma = \mathbf{s}_1 - \mathbf{s}_2 \in \mathcal{S}_{[t_1,t_2]}.$$

Therefore, $(\mathbf{x}_2 - \mathbf{x}_1, \sigma)$ is a legitimate message-signature of the KKS scheme.

Trapdoor collision: One who has the trapdoor, *i.e.,* the pair $(\mathbf{G}, J)$, can find a collision as above.

Uniformity: Define

$$\begin{aligned}D_1 &= \left\{ f(\mathsf{ppk}, \mathbf{x}_1, \mathbf{s}_1) \mid \mathbf{s}_1 \leftarrow \mathcal{S}_t \right\},\\ D_2 &= \left\{ f(\mathsf{ppk}, \mathbf{x}_2, \mathbf{s}_2) \mid \mathbf{s}_2 \leftarrow \mathcal{S}_{[t-t_2,t+t_2]} \right\}.\end{aligned}$$

Using hybrid arguments, we show that D_1 and D_2 are computationally indistinguishable as follows.

1. For $i = 1, 2$, let $\mathcal{U}_i$ be the induced distribution of $f(\mathsf{ppk}, \mathbf{x}_i, \mathbf{s})$ over $\mathbb{F}_q^{N-K}$, where $\mathbf{s} \leftarrow \mathbb{F}_q^N$. Since $\mathbf{H}$ is of full rank, thus for a random vector $\mathbf{v} \in \mathbb{F}_q^{N-K}$,

$$\begin{aligned}\Pr_{\mathbf{s}\leftarrow\mathbb{F}_q^N}\left[f(\mathsf{ppk}, \mathbf{x}_1, \mathbf{s}) = \mathbf{v}^T\right] &= \Pr_{\mathbf{s}\leftarrow\mathbb{F}_q^N}\left[\mathbf{H}\cdot\mathbf{s}^T = \mathbf{v} - \mathbf{F}\cdot\mathbf{x}_1^T\right]\\ &= \frac{q^{N-K}}{q^N}\\ &= q^{-K}.\end{aligned}$$

   The same equality holds true when $\mathbf{x}_1$ is replaced by $\mathbf{x}_2$. Therefore, we have that $\mathcal{U}_1 \equiv \mathcal{U}_2$. (The parameters are chosen such that $K = N - K$, therefore, $\mathcal{U}_1$ and $\mathcal{U}_2$ are identical with the uniform distribution over $\mathbb{F}_q^{N-K}$.)

2. By the hardness of $\mathsf{DSD}(N, K, t)$ problem, D_1 and $\mathcal{U}_1$ are computationally indistinguishable.

3. By the hardness of $\mathsf{extDSD}(N, K, t - t_2, t + t_2)$ problem, D_2 and $\mathcal{U}_2$ are computationally indistinguishable.

Consequently, we conclude that D_1 and D_2 are computationally indistinguishable. □

3.4 A Signature Scheme using f

In this section, we describe a signature scheme constructed from the function f. We follow the methodology from [BKKP15]: first, we show how f can be used to build a one-time two-tier signature scheme, then in a black-box manner we move from the one-time two-tier construction to a non-adaptive signature scheme, and with an extra use of f, we can obtain a regular signature scheme in the standard model.

3.4.1 A One-time Two-tier Scheme

Our first step is to establish a one-time two-tier signature scheme, called TTSig_f. The descriptions of f, $\mathcal{S}_t$, and $\mathcal{S}_{[t-t_2,t+t_2]}$ are as in Section 3.3. The message space is $\mathbb{F}_q^k$.

- $\mathsf{PriGen}(1^\lambda)$: Use the setting procedure of the KKS scheme. The primary public key is $\mathsf{ppk} = (\mathbf{H}, \mathbf{F})$ and the primary secret key is $\mathsf{psk} = (\mathbf{G}, J)$.
- $\mathsf{SecGen}(\mathsf{ppk}, \mathsf{psk})$: Choose $\widehat{\mathbf{s}} \leftarrow \mathcal{S}_t$, and compute $h = \mathsf{CHash}(\mathsf{ppk}, \widehat{\mathbf{x}}, \widehat{\mathbf{s}})$ for some arbitrary public $\widehat{\mathbf{x}} \in \mathbb{F}_q^k$. The secondary public key is $\mathsf{spk} = h$, and the secondary secret key is $\mathsf{ssk} = \widehat{\mathbf{s}}$.
- $\mathsf{TTSign}(\mathsf{psk}, \mathsf{ssk}, \mathbf{x})$: The signer uses trapdoor $(\mathbf{G}, J)$ to compute a collision as $\mathbf{s} = \mathsf{Coll}(\mathsf{psk}, \widehat{\mathbf{x}}, \widehat{\mathbf{s}}, \mathbf{x})$. The signature on $\mathbf{x}$ is $\mathbf{s} \in \mathcal{S}_{[t-t_2,t+t_2]}$.
- $\mathsf{TTVerify}(\mathsf{ppk}, \mathsf{spk}, \mathbf{x}, \mathbf{s})$: The verifier checks the condition $\mathsf{CHash}(\mathsf{ppk}, \mathbf{x}, \mathbf{s}) = h$.

The security of the scheme is stated in the following theorem.

Theorem 3.1. *If f is a (t, ε)-collision resistant chameleon hash function, then for any positive integer q, TTSig_f is a $(t', q, 1, \varepsilon')$-TT-EUF-NCMA signature, where $\varepsilon' = \varepsilon$, and $t' = t - O(q)$.*

Proof. Let $\mathcal{F}$ be a probabilistic polynomial-time adversary that $(t', q, 1, \varepsilon')$-breaks the TT-EUF-NCMA security of TTSig_f. Then we construct an adversary $\mathcal{B}$ that (t, ε)-breaks the collision resistance of f. Formally, $\mathcal{B}$ is given the challenge

chameleon hash key chk and asked to come up with two pairs $(\mathbf{x}, \mathbf{s}), (\mathbf{x}', \mathbf{s}')$ such that $\mathbf{x} \neq \mathbf{x}'$ and $\mathsf{CHash}(\mathsf{chk}, \mathbf{x}, \mathbf{s}) = \mathsf{CHash}(\mathsf{chk}, \mathbf{x}', \mathbf{s}')$.

Simulation. $\mathcal{B}$ simulates $\mathsf{PriGen}(1^\lambda)$ as follows: it sets $\mathsf{ppk} = \mathsf{chk}$ and returns ppk to $\mathcal{F}$. Now, $\mathcal{B}$ does not have the chameleon hash trapdoor and psk is empty. Upon receiving the i-th message $\mathbf{x}_i$ from $\mathcal{F}$, $\mathcal{B}$ simulates $\mathtt{NTTSign}(\mathbf{x}_i)$ as follows: it picks a random $\mathbf{s}_i \leftarrow \mathcal{S}_{[t-t_2,t+t_2]}$ and computes $h_i = \mathsf{CHash}(\mathsf{ppk}, \mathbf{x}_i, \mathbf{s}_i)$; defines the secondary public key $\mathsf{spk}_i = h_i$ and returns spk_i and the signature $\mathbf{s}_i$. The simulation is computationally indistinguishable from the real execution. Firstly, chk is from the chameleon hash challenge and, thus, the simulation of PriGen is identical to the definition. Secondly, in the original definition $\mathsf{spk}_i = \mathsf{CHash}(\mathsf{ppk}, \mathbf{0}, \mathbf{r}_i)$, where $\mathbf{r}_i \leftarrow \mathcal{S}_t$ and $\mathsf{spk}_i = \mathsf{CHash}(\mathsf{ppk}, \mathbf{x}_i, \mathbf{s}_i)$ in the simulation. These two distributions are computationally indistinguishable based on the uniformity of f. Thirdly, it is easy to see the simulated signatures are well-formed.

Extracting collision. Once $\mathcal{F}$ outputs a forgery $(\mathbf{x}^*, \mathbf{s}^*, i^*)$, $\mathcal{B}$ aborts if spk_{i^*} is undefined. Otherwise, $\mathcal{B}$ checks if

$$\mathsf{CHash}(\mathsf{ppk}, \mathbf{x}_{i^*}, \mathbf{s}_{i^*}) = \mathsf{spk}_{i^*} = \mathsf{CHash}(\mathsf{ppk}, \mathbf{x}^*, \mathbf{s}^*).$$

If that is the case, then $\mathcal{B}$ returns the collision $\big((\mathbf{x}^*, \mathbf{s}^*), (\mathbf{x}_{i^*}, \mathbf{s}_{i^*})\big)$. By the unforgeability of TTSig_f, we have $\mathbf{x}^* \neq \mathbf{x}_{i^*}$. Thus, if $\mathcal{F}$ outputs a successful forgery, then $\mathcal{B}$ finds a collision for the chameleon hash with probability $\varepsilon = \varepsilon'$.

□

3.4.2 A Non-adaptive Signature Scheme

By adapting the generic constructions from [BKKP15] for the above one-time two-tier scheme, we immediately obtain a stateful scheme $\mathsf{BinTree}[\mathsf{TTSign}] = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ using a binary tree of height ℓ where we assume the message space to be of size 2^ℓ.

The signer will implicitly hold a binary tree of depth ℓ. Every node $v \in \{0,1\}^{\leq \ell}$ has a label L_v which is a secondary public key of the two-tier scheme. All nodes can be computed "on the fly." Each leaf is used to sign a single message. When signing message m, the signer takes the leftmost unused leaf $v_\ell \in \{0,1\}^\ell$ in the tree and generates the label $L_{v_\ell} \leftarrow \mathsf{SecGen}(\mathsf{ppk}, \mathsf{psk})$. Define $L_{v_{\ell+1}} = m$. Then the path from the root v_0 to v_ℓ is computed. For each undefined node v_i on the path, the signer assigns label $L_{v_i} \leftarrow \mathsf{SecGen}(\mathsf{ppk}, \mathsf{psk})$. After that, every node on the path is signed using the label of its parent.

When signing the nodes on the path, the signer takes the node v_i in the top-down manner and signs both children of v_i under L_{v_i},

$$\sigma_{i+1} \leftarrow \mathsf{Sign}(\mathsf{psk}, \mathsf{ssk}_{v_i}, \mathsf{Child}_l || \mathsf{Child}_r),$$

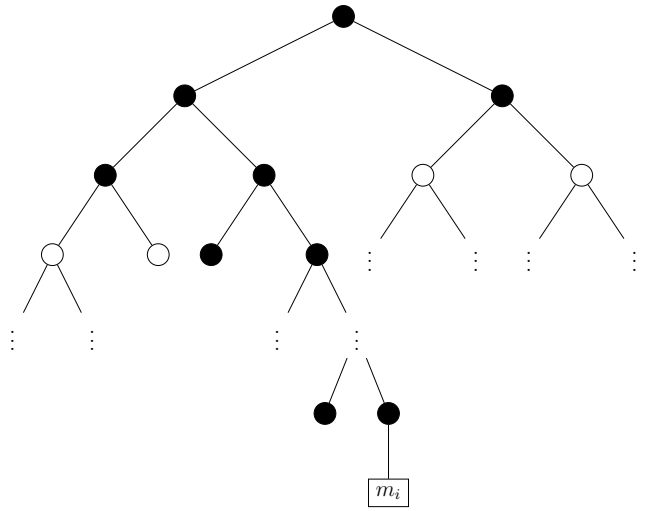

Figure 3.1: Nodes in black are used in the i-th Signature with BinTree[TTSign].

where ssk_{v_i} is the secondary secret key associated with node v_i, and Child_l and Child_r are the left and right children of node v_i, respectively. The signer outputs the path and the two-tier signatures on the path as the signature of m.

3.4.3 Wrapping-up

To obtain a signature scheme, one can now use the classical transformation from [KR00] using an extra chameleon hash, so another use of the function f on the leaves. In other words, we pick a fresh $\mathbf{s} \leftarrow \mathcal{S}_t$, set $L_{v_{\ell+1}} = f(\mathsf{ppk}, m, \mathbf{s})$, proceed as before to build the tree (using $L_{v_{\ell+1}}$ as the target message), and output both $\mathbf{s}$ and this non-adaptive signature. The resulting signature is stateful. Depending on the applications, one might prefer to move to a stateless scheme, once again, there exist generic techniques, like the one presented by Goldreich in [Gol87], where basically every randomness is generated through a pseudo-random function. However, as here, we need to keep only the ℓ *active* nodes, there is not a huge blow up in memory for the signer, so it is not really worth the trade-off.

3.5 Parameters

We have chosen the parameters of the scheme such that the best practical information set decoding algorithm has complexity at least 2^λ for solving $\mathsf{DSD}(N, K, t)$, $\mathsf{extDSD}(N, K, t - t_2, t + t_2)$ or breaking the KKS assumption.

To obtain the security level $\lambda = 128$, we chose

$$N = 48948, K = N/2, n = 892, k = 43, t_1 = 219, t_2 = 674, t = 893, q = 2.$$

By Proposition 3.2, the probability that signatures (of the KKS scheme) do not lies in the interval $[t_1, t_2]$ is at most $2^{-133.4}$.

The size of pk and the signature are given as follows:

- The public key is $\mathsf{pk} = (\mathsf{ppk}, \mathsf{spk}_\varepsilon)$, where $\mathsf{ppk} = (\mathbf{F}, \mathbf{H})$, and $\mathsf{spk}_\varepsilon = \mathsf{SecGen}(\mathsf{ppk}, \mathsf{psk})$. The size of pk is dominated by the size of $\mathsf{ppk} = (\mathbf{H}, \mathbf{F})$. Note that the matrix $\mathbf{H}$ is of size $(N-K) \times N$, the matrix $\mathbf{F} = \mathbf{H}_J \cdot \mathbf{G}^T$ and thus is of size $(N-K) \times k$, hence the size of ppk is $(N+k) \cdot (N-K)$.

- If one wants to be able to sign 2^ℓ times, then the signature is of the form

$$\sigma = \big(v_h, (L_{v_0}, \sigma_0), \ldots, (L_{v_\ell}, \sigma_\ell), (L_{v_{\ell+1}}, \mathbf{s})\big),$$

  where each $L_{v_i}, 0 \le i \le \ell$ is an output of $\mathsf{SecGen}(\mathsf{ppk}, \mathsf{psk})$ and thus is a vector in $\mathbb{F}_2^{N-K}$, and hence of size about $N-K$; the size of $L_{v_{\ell+1}} = f(\mathsf{ppk}, m, \mathbf{s}) \in \mathbb{F}_q^{N-K}$ is $N-K$. Each $\sigma_*, \mathbf{s}$ is a chameleon hash opening so of size N; v_h is the node in used and of size ℓ. Hence, the size of σ is $\ell + (2N-K) \cdot (\ell+1) + N + N - K = \ell + (\ell+2) \cdot (2N-K) = O(\ell N)$.

Table 3.2 provides some examples of parameters for a security parameter $\lambda = 128$.

ℓ	Size of public key $\lvert\mathsf{pk}\rvert$ (bytes)	Size of signatures $\lvert\sigma\rvert$ (bytes)
4	$2^{27.2}$	56080
8	$2^{27.2}$	93466
12	$2^{27.2}$	130853

Table 3.2: Examples of parameters for the tree-based scheme.

3.6 Some Observations

The key size and signature size of the scheme are quite large. This is because those of the underlying KKS scheme are comparatively large also.

There might be two possible ways to overcome this disadvantage that one can think of.

(*i*) As mentioned in [OT11], when choosing parameters, the ratio $\Re = \frac{k}{N}$ should be significantly greater than $\mathfrak{r} = \frac{k}{n}$. In this work, $\Re \approx 10.37\mathfrak{r}$, and we did not make the attempt to optimize this ratio. Thus, one could try to decrease this ratio and find the best trade-off between parameters and security.

(*ii*) Another direction could be carrying out the entire construction to rank metric. It is often the case that with the same scheme, rank metric provides smaller size of objects than those in the HAMMING metric.

We do not dive in detail here and leave these speculations for the future works.

Chapter 4

Group Signatures in the Rank Metric

The second contribution is a code-based group signature scheme in the rank metric context. Following the same track as that of Ezerman *et al.* [ELL+15], still, our scheme differs from the previous by giving another solution to the same problem. The solution of [ELL+15] makes use of the special structure of $\mathbb{F}_2$, *i.e.,* this field has only two elements 0 and 1, and thus would lose its elegant or even become inefficient when being applied to other fields. Because of this limitation, this method clearly could not be used to derive the rank version. On the other hand, our solution does not rely on any specific field structure like the previous solution. The main purpose of this solution is for the construction of a scheme in the rank metric; however, it can also be feasible for the Hamming case. In general, the parameters of our scheme are quite in the same level as those of [ELL+15], and in some cases ours are better.

This is a joint work with Olivier Blazy and Philippe Gaborit and appeared in CBCrypto 2021.

4.1 Introduction

Designing group signature is one of the most intriguing problem in cryptography. The ultimate goal is schemes which satisfy the fundamental requirements of a group signature scheme, and meet practicable purposes. Especially that the era of quantum computing is coming, which would make number-theoretic based group signature schemes insecure, the search for post-quantum schemes has become active than ever. Much of proposals are published in both lattice-based assumptions and code-based assumptions.

On the lattice-based side, there have schemes such as [LLNW14, NZZ15]. The-

oretically, these schemes provide efficient public key size and signature size in the asymptotic sense. (The size of signature is only linear in logarithm of the number of users.) However, as pointed out in [ELL+15], when being instantiated with practical parameters, they suffer from large key and signature sizes.

On the code-based side, there have constructions on both static and dynamic group, *e.g.,* [ABCG16a, ABCG16b, ELL+15]. The scheme presented in [ABCG16b] used the RankSign primitive. However, RankSign signature scheme was broken in [DT18]. Thus, actually, there is no group signature scheme based on rank metric. The first static code-based group signature scheme in HAMMING metric was designed by EZERMAN *et al.* [ELL+15]. This scheme provides public key and signature sizes being linear in the number of users which really is a weak point compared to those of lattice-based. Despite this fact, at the same level of security ($\lambda = 80$), their parameters are remarkably smaller than those of lattice schemes. Take a closer look, their construction uses 3 cryptographic layers:

1. The first layer is a signature scheme derived from STERN's identification protocol through FIAT-SHAMIR transform.

2. The second layer is the randomized MCELIECE encryption scheme which is used to encrypt identity of the signer.

3. The third layer is a zero-knowledge (ZK) protocol that links the two above layers together. It allows one to show that a given signature is generated by a certain user in the group who honestly encrypts his identity information.

It was emphasized also in their paper that "Constructing such protocol is quite challenging."

Our contribution. In this work, we revisit that challenge and show how to adapt it in the rank context. Since [ABCG16b] is broken, our scheme becomes the first rank metric group signature scheme, which, moreover, relies on generic problems. When being instantiated with concrete parameters at the same security level $\lambda = 128$, the size of signatures of our scheme are smaller than those of [ELL+15]; and when the values of ℓ are not too large, *e.g.,* $\ell = 4, 8, 12$, the size of public keys are also less than those of [ELL+15]. Our parameters are set as in Section 4.5. For the schemes of [ELL+15] to attain security level 128, we take $(n, k, t) = (2^{12}, 3604, 41)$ as in [FS09], and for the syndrome decoding problem (Problem 1), we try to set $(m, r, w) = (4097, 721, 162)$ so that it also satisfies Lemma 1 in [ELL+15].

Overview of Our Techniques. Let $k, \ell, m, m_0, n, n_0, r_0, w_r, w_s$ be positive integers. We consider a group of $N-1 = q^\ell - 1$ users, where q is a power of a prime number. (The reason for this way of denoting will be clear in the sequel.) Each user is indexed by an integer $j \in \{1, \ldots, N-1\}$ and has a signing key $\mathbf{s}_j$, which

ℓ	PK size		Signature size	
	Our scheme	[ELL+15]	Our scheme	[ELL+15]
4	16.71 KB	2.22 MB	2.94 MB	3.05 MB
8	77.68 KB	2.24 MB	2.95 MB	3.05 MB
12	1.05 MB	2.58 MB	3.06 MB	3.12 MB
16	16.57 MB	8.12 MB	4.74 MB	4.85 MB

Table 4.1: Comparison with [ELL+15].

is randomly chosen from $\mathcal{S}_{w_s}^{n_0,m_0}$, *i.e.,* the set of vectors of rank wight w_s in $\mathbb{F}_{q^{m_0}}^{n_0}$. A part of the public key contains a matrix $\mathbf{H} \in \mathbb{F}_{q^{m_0}}^{r_0 \times n_0}$, which is the parity matrix in the systematic form of an ideal code, and $N-1$ syndromes $\mathbf{y}_1, \ldots, \mathbf{y}_{N-1} \in \mathbb{F}_{q^m}^{r_0}$ such that $\mathbf{H} \cdot \mathbf{s}_j^T = \mathbf{y}_j^T$. Our three layers are as follows.

1. **The signature layer.** Let $\mathbf{A} = \left[\mathbf{y}_1^T | \cdots | \mathbf{y}_{N-1}^T\right] \in \mathbb{F}_{q^{m_0}}^{r_0 \times (N-1)}$, and $\mathbf{x} = \delta_j^{N-1}$ - the vector of dimension $N-1$ with 1 at the jth position and 0 elsewhere. In this layer, the user uses Stern's framework in the rank context to prove that he possesses a pair $(\mathbf{s}, \mathbf{x})$ satisfying

$$\mathbf{H} \cdot \mathbf{s}^T - \mathbf{A} \cdot \mathbf{x}^T = \mathbf{0}. \tag{4.1}$$

   Then, the protocol is transformed into a Fiat-Shamir signature.

2. **The encryption layer.** We use RQC encryption scheme to encrypt the identity information of the users. Each index $j \in \{1, \ldots, N-1\}$ is mapped to a vector in $\mathbb{F}_{q^m}^k$, the message space of an RQC scheme, by function $\mathsf{I2V}(\cdot)$. The ciphertext is of the form $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ such that

$$\begin{cases} \mathbf{c}_1 = \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2, \\ \mathbf{c}_2 = \mathbf{r}_3 + \mathbf{s} \cdot \mathbf{r}_2 + \mathsf{I2V}(j) \cdot \mathbf{G}, \end{cases} \tag{4.2}$$

   where $\mathbf{h}, \mathbf{s} \in \mathbb{F}_{q^m}^n$ is the public key, $\mathbf{G} \in \mathbb{F}_{q^m}^k$ is a generator matrix of a public code, and $\mathbf{r}_i \in \mathcal{S}_{w_r}^{n,m}$ for $i = 1, 2, 3$.

3. **The third layer.** This ingredient is a ZK protocol that allows the user to show that the vector $\mathbf{x} = \delta_j^{N-1}$ used in the first layer and the hidden plaintext $\mathsf{I2V}(j)$ used in the second layer both point to the same $j \in \{1, \ldots, N-1\}$. Our solution to this problem is as follows. Let $f_1 \colon \{1, \ldots, N-1\} \to \mathbb{F}_q^\ell$ be the function that maps each element of $\{1, \ldots, N-1\}$ to a different element of $\mathbb{F}_q^\ell$, $f_2 \colon \mathbb{F}_q^\ell \to \mathbb{F}_{q^m}^k$ be a function that map each vector in $\mathbb{F}_q^\ell$ to a vector in $\mathbb{F}_{q^m}^k$; their inverses are denoted by f_1^{-1}, f_2^{-1}, respectively. The map $\mathsf{I2V}$

is defined as $\mathsf{I2V}(j) = f_2 \circ f_1(j)$. For every $\mathbf{v} \in \mathbb{F}_q^\ell \setminus \{\mathbf{0}\}$, we construct two permutations $P_\mathbf{v} \colon \mathbb{F}_q^{N-1} \to \mathbb{F}_q^{N-1}$ and $P'_\mathbf{v} \colon S \to S$, where $S = \big\{\mathsf{I2V}(j) \mid j \in \{1, \ldots, N-1\}\big\} \cup \{\mathbf{0}\}$, such that for any $j \in \{1, \ldots, N-1\}$ we have

$$\begin{aligned} \mathbf{x} = \delta_j^{N-1} &\Longleftrightarrow P_\mathbf{v}(\mathbf{x}) = \delta^{N-1}_{f_1^{-1}\left(f_1(j)\cdot\mathbf{v}\right)}, \\ \mathbf{m} = \mathsf{I2V}(j) &\Longleftrightarrow P'_\mathbf{v}(\mathbf{m}) = \mathsf{I2V}\left(f_1^{-1}\big(f_1(j)\cdot\mathbf{v}\big)\right). \end{aligned}$$

In the protocol, the user randomly picks a non zero vector $\mathbf{v} \in \mathbb{F}_q^\ell \setminus \{\mathbf{0}\}$ and sends $\mathbf{v}' = f_1(j) \cdot \mathbf{v}$. The verifier, seeing that $P_\mathbf{v}(\mathbf{x}) = \delta^{N-1}_{f_1^{-1}(\mathbf{v}')}$ and $P'_\mathbf{v}(\mathbf{m}) = \mathsf{I2V}\big(f_1^{-1}(\mathbf{v}')\big)$, should be convinced that $\mathbf{x}$ and $\mathbf{m}$ link to the same $j \in \{1, \ldots, N-1\}$, yet the value of j is completely hidden. Here, vector $\mathbf{v}$ also acts as a one-time pad as in the case of addition, *i.e.,* the operation $\oplus / +$. We remark that our method can be applied very well in the case of [ELL+15]. Note that the technique used in [ELL+15] heavily relies on the particular value $q = 2$. When the binary field is replaced by any finite field, this technique would lose its efficiency.

With this technique embedded in Stern's framework, the user can convince the verifier that he possesses a tuple $(j, \mathbf{s}, \mathbf{x}, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ satisfying 4.1 and 4.2. By repeating this protocol many times, and making non-interactive, we get a ZK proof of knowledge Π. The final signature is of the form $(\mathbf{c}, \Pi)$. In the random oracle model, the anonymity of the scheme relies on the zero-knowledge of Π and the CPA-security of the RQC; its traceability relies on the hardness of Rank Syndrome Decoding problem.

4.2 Preliminaries

4.2.1 Notations

We use bold low-case letters to denote vectors and bold capital letters for matrices. The transpose of a vector $\mathbf{x}$ is denoted by $\mathbf{x}^T$. The same notation is used for matrices. The rank weight of a vector $\mathbf{x}$ is denoted by $\|\mathbf{x}\|$. The set of invertible matrices of size m over $\mathbb{F}_q$ is denoted by $\mathrm{GL}(m, q)$. For a positive integer $N > 1$, $[N-1] \stackrel{\text{def}}{=} \{1, \ldots, N-1\}$. The sphere of radius r centered at $\mathbf{0}$ in $\mathbb{F}_{q^m}^n$ is denoted by $\mathcal{S}_r^{n,m}$. By writing $x \leftarrow X$, we mean that x is drawn according to the distribution X, if X is a distribution; or drawn uniformly at random from X when X is a set; or output of the algorithm X, if X is an algorithm.

4.2.2 Background on Code-Based Cryptography

The basic notions on rank metric are already provided in Section 2.2. Now, let $f(X) \in \mathbb{F}_{q^m}[X]$ be a polynomial of degree n and $\mathbb{F}_{q^m}[X]/\langle f \rangle = \mathcal{R}_f$. Consider the following map:

$$\begin{aligned} \phi\colon \mathbb{F}_{q^m}^n &\longrightarrow \mathcal{R}_f \\ (a_0, \ldots, a_{n-1}) &\longmapsto a_0 + \cdots + a_{n-1}X^{n-1}. \end{aligned}$$

The inverse map, denoted by ϕ^{-1}, simply maps a polynomial to the vector formed by its coefficients. For the sake of simplicity, if $\mathbf{a} = (a_0, \ldots, a_{n-1}) \in \mathbb{F}_{q^m}^n$, we let $\phi(\mathbf{a}) = a_0 + \cdots + a_{n-1}X^{n-1} = a(X)$. For $\mathbf{a}, \mathbf{b} \in \mathbb{F}_{q^m}^n$, their product $\mathbf{a} \cdot \mathbf{b}$ is defined as

$$\mathbf{a} \cdot \mathbf{b} = \phi^{-1}\big(a(X) \cdot b(X)\big).$$

Clearly, we have $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$. It is also not hard to see that

$$\mathbf{a} \cdot \mathbf{b} = (a_0, \ldots, a_{n-1}) \cdot \begin{pmatrix} \phi^{-1}\big(b(X)\big) \\ \vdots \\ \phi^{-1}\big(X^{n-1}b(X)\big) \end{pmatrix}. \tag{4.3}$$

Consider the case when $f(X)$ is irreducible over $\mathbb{F}_{q^m}$, then $g(X)$ and $f(X)$ are coprime for any nonzero $g(X) \in \mathcal{R}_f$. Thus, for an arbitrary nonzero $g(X) \in \mathcal{R}_f$, if we define

$$g \cdot \mathcal{R}_f = \big\{ g(X) \cdot a(X) \mod f \mid a(X) \in \mathcal{R}_f \big\},$$

then we have $g \cdot \mathcal{R}_f = \mathcal{R}_f$. From this observation, we deduce that $\mathbf{g} \cdot \mathbb{F}_{q^m}^n = \mathbb{F}_{q^m}^n$ (here, $\mathbf{g} = \phi^{-1}(g) \neq \mathbf{0}$), that is to say, $\mathbf{g}$ together with the multiplication operation defined above form a permutation over $\mathbb{F}_{q^m}^n$. (This permutation fixes $\mathbf{0}$.) For our purpose, it is enough to consider $m = 1$ and $f(X)$ is irreducible over $\mathbb{F}_q$.

The right-most term on the right hand side of Equation 4.3 is usually referred to as the ideal matrix generated by $b(X)$ with respect to $f(X)$. For ease of notation, vectors are identical with their corresponding polynomials, *i.e.,* $X^k\mathbf{b}$ is understood to be $\phi^{-1}\big(X^k b(X)\big)$. Thus, the ideal matrix of a vector $\mathbf{b}$ with respect to f is written as

$$\mathfrak{b} = \begin{pmatrix} \mathbf{b} \\ X \cdot \mathbf{b} \\ \vdots \\ X^{n-1} \cdot \mathbf{b} \end{pmatrix}.$$

In our construction, we will use 2-ideal codes and 3-ideal codes. A 2-ideal code of length $2n$ over $\mathbb{F}_{q^m}$ is a code whose parity matrix is of the form

$$\mathbf{H} = \big[\mathbf{I}_n \mid \mathfrak{b}^T\big], \tag{4.4}$$

where $\mathfrak{h}$ is the ideal matrix of a vector $\mathbf{h}$ in $\mathbb{F}_{q^m}^n$. Similarly, a 3-ideal code of length $3n$ over $\mathbb{F}_{q^m}$ is a code whose parity matrix is of the form

$$\mathbf{H} = \begin{pmatrix} \mathbf{I}_n & \mathbf{0} & \mathfrak{h}_1^T \\ \mathbf{0} & \mathbf{I}_n & \mathfrak{h}_2^T \end{pmatrix}. \tag{4.5}$$

Next, we recall some definitions concerning code-based hardness assumptions related to this type of codes. These are, in fact, particular cases of Problem 4 and Problem 5 in which completely random matrices are replaced by ideal matrices. In the following definitions, $\nu \in \{2, 3\}$ and $S(n, \nu)$ is the set of all matrices of the form as in Equation 4.4 or 4.5 corresponding to the case $\nu = 2$ or $\nu = 3$, respectively.

Definition 4.1 ($\nu - \mathsf{IRSD}$ Distribution)**.** *Let n, w be positive integers, $P(X) \in \mathbb{F}_q[X]$ be an irreducible polynomial of degree n. The $\nu - \mathsf{IRSD}(n, w)$ distribution chooses uniformly at random a matrix $\mathbf{H} \in S(n, \nu)$ together with a vector $\mathbf{x} \in \mathbb{F}_{q^m}^{\nu n}$ such that $\|\mathbf{x}\| = w$ and outputs $\left(\mathbf{H}, \mathbf{H} \cdot \mathbf{x}^T\right)$.*

Definition 4.2 (Computational $\nu - \mathsf{IRSD}$ Problem)**.** *Let n, w be positive integers, $P(X) \in \mathbb{F}_q[X]$ be an irreducible polynomial of degree n, $\mathbf{H} \in S(n, \nu)$ be a random matrix, and $\mathbf{y} \leftarrow \mathbb{F}_{q^m}^n$. The computational $\nu - \mathsf{IRSD}(n, w)$ problem asks to find a vector $\mathbf{x} \in \mathbb{F}_{q^m}^{\nu n}$ such that $\|\mathbf{x}\| = w$ and $\mathbf{H} \cdot \mathbf{x}^T = \mathbf{y}^T$.*

Definition 4.3 (Decisional $\nu - \mathsf{IRSD}$ Problem)**.** *The decisional $\nu - \mathsf{IRSD}(n, w)$ problem asks to decide with non-negligible advantage whether $(\mathbf{H}, \mathbf{y}^T)$ came from the $\nu - \mathsf{IRSD}(n, w)$ distribution or the uniform distribution over $S(n, \nu) \times \mathbb{F}_{q^m}^n$.*

The RQC scheme. In the Encryption layer, we make use of the RQC scheme [ABD+16]. It is as follows.

- $\mathsf{RQC.Setup}(1^\lambda)$: Generate parameters $m = m(\lambda), n = n(\lambda), k = k(\lambda), w_r = w_r(\lambda)$, an irreducible polynomial $P[X] \in \mathbb{F}_q[X]$, which is also irreducible in $\mathbb{F}_{q^m}[X]$. The plaintext space is $\mathbb{F}_{q^m}^k$. Output $\mathsf{param} = (m, n, k, w_r, P)$.
- $\mathsf{RQC.KeyGen}(\mathsf{param})$: Generate $\mathbf{h} \leftarrow \mathbb{F}_{q^m}^n, \mathbf{x}, \mathbf{y} \leftarrow \mathcal{S}_{w_r}^{n,m}$ sharing the same support, a generator matrix $\mathbf{G} \in \mathbb{F}_{q^m}^{k \times n}$ of a public code $\mathcal{C}$. Output $\mathsf{pk_{RQC}} = (\mathbf{h}, \mathbf{s} = \mathbf{x} + \mathbf{h} \cdot \mathbf{y}, \mathbf{G})$ and $\mathsf{sk_{RQC}} = (\mathbf{x}, \mathbf{y})$.
- $\mathsf{RQC.Enc}(\mathsf{pk_{RQC}}, \mathbf{m})$: To encrypt a message $\mathbf{m} \in \mathbb{F}_{q^m}^k$, choose $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3 \leftarrow \mathcal{S}_{w_r}^{n,m}$, which belong to the same support. Compute

$$\begin{cases} \mathbf{c}_1 = \mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2, \\ \mathbf{c}_2 = \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{r}_3 + \mathbf{m} \cdot \mathbf{G}. \end{cases}$$

- $\mathsf{RQC.Dec}(\mathsf{sk}_{\mathsf{RQC}}, \mathbf{c})$: Apply the decoding algorithm of the code $\mathcal{C}$ to

$$\mathbf{y} \cdot \mathbf{c}_1 - \mathbf{c}_2 = \mathbf{x} \cdot \mathbf{r}_2 - \mathbf{y} \cdot \mathbf{r}_1 + \mathbf{r}_3 + \mathbf{m} \cdot \mathbf{G}.$$

For the sake of convenience, define

$$\mathbf{H}_1 = \begin{pmatrix} \mathbf{I}_n \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{H}_2 = \begin{pmatrix} \mathfrak{h}^T \\ \mathfrak{g}^T \end{pmatrix}, \quad \mathbf{H}_3 = \begin{pmatrix} \mathbf{0} \\ \mathbf{I}_n \end{pmatrix}, \quad \mathbf{H}_4 = \begin{pmatrix} \mathbf{0} \\ \mathbf{G}^T \end{pmatrix},$$

then we have $\left[\mathbf{H}_1 | \cdots | \mathbf{H}_4\right] \cdot (\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{m})^T = \left(\mathbf{c}_1, \mathbf{c}_2\right)^T$.

The RQC scheme is CPA-secure; its security relies on the hardness of the decisional $2 - \mathsf{IRSD}(n, w_r)$ and $3 - \mathsf{IRSD}(n, w_r)$ problems as has been proven in [ABD+16]. (Although the proof therein is applied for quasi-cyclic codes, a proof for ideal codes can be derived straightforwardly.)

4.2.3 Group Signatures

In this section, we recall some definitions of group signatures following [BMW03] on the case of static groups.

Definition 4.4. *A group signature scheme* $\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ *contains four polynomial-time algorithms:*

1. KeyGen *takes as input* $(1^\lambda, 1^N)$, *where* λ *is the security parameter and* N *is a positive integer which is the number of group users, and returns a tuple* $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$, *where* gpk *is the group public key,* gmsk *is the group manager's secret key, and* $\mathsf{gsk} = \{\mathsf{gsk}[j]\}_{j \in [N-1]}$ *with* $\mathsf{gsk}[j]$ *being the secret key of the group user of index* j.

2. Sign *takes as input a message* M, *a secret key* $\mathsf{gsk}[j]$ *in the set* gsk *and returns a group signature* Σ *on* M.

3. Verify *takes as input the group public key* gpk, *a message* M, *a signature* Σ *on* M, *and returns either* 1 (Accept) *or* 0 (Reject).

4. Open *takes as input the group manager's secret key* gmsk, *a signature* M, *a signature* Σ *on* M, *and returns an identity* j *or the symbol* $\perp$ *to indicate failure.*

Correctness: The correctness of a group signature scheme requires that for all positive integers λ, N, all output $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk})$ of KeyGen, all identity j, and all message $M \in \{0,1\}^*$,

$$\begin{cases} \mathsf{Verify}(\mathsf{gpk}, M, \mathsf{Sign}(\mathsf{gsk}[j], M)) = 1, \\ \mathsf{Open}(\mathsf{gmsk}, M, \mathsf{Sign}(\mathsf{gsk}[j], M)) = j. \end{cases}$$

Security Notions: A secure group signature scheme must satisfy two security requirements:

1. *Traceability* requires that all signatures can be traced back to the identity of its signer, even in the case there is a collusion between the group users.

2. *Anonymity* requires that signatures generated by two users are computationally indistinguishable to an adversary who knows all the secret keys.

We follow [ELL+15] by stating the security definitions.

Definition 4.5. *A group signature scheme* $\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ *is* CPA*-anonymous if for all polynomial* $N(\cdot)$ *(in* λ*) and any probabilistic polynomial-time adversary* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in the following experiment is negligible in* λ*:*

1. *Run* $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^N)$ *and send* $(\mathsf{gpk}, \mathsf{gsk})$ *to* $\mathcal{A}$*.*

2. $\mathcal{A}$ *outputs two identities* $j_0, j_1 \in [N-1]$ *together with a message* M*. Choose a random bit* b *and give* $\mathsf{Sign}(\mathsf{gsk}[j_b], M)$ *to* $\mathcal{A}$*. Then* $\mathcal{A}$ *outputs a bit* b'*.*

$\mathcal{A}$ *succeeds if* $b' = b$*. The advantage of* $\mathcal{A}$ *is defined to equal* $\left|\Pr[\mathcal{A} \text{ succeeds}] - \frac{1}{2}\right|$*.*

Definition 4.6. *A group signature* $\mathcal{GS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify}, \mathsf{Open})$ *is traceable if for all polynomial* $N(\cdot)$ *and any adversary* $\mathcal{A}$*, the success probability of* $\mathcal{A}$ *in the following experiment is negligible in* λ*:*

1. *Run* $(\mathsf{gpk}, \mathsf{gmsk}, \mathsf{gsk}) \leftarrow \mathsf{KeyGen}(1^\lambda, 1^N)$ *and send* $(\mathsf{gpk}, \mathsf{gsk})$ *to* $\mathcal{A}$*.*

2. $\mathcal{A}$ *may query the following oracles adaptively and in any order:*

   - *An* $\mathcal{O}^{\mathsf{Corrupt}}$ *oracle that on input* $j \in [N-1]$*, outputs* $\mathsf{gsk}[j]$*.*
   - *An* $\mathcal{O}^{\mathsf{Sign}}$ *oracle that on input* j *and a message* M*, returns* $\mathsf{Sign}(\mathsf{gsk}[j], M)$*.*

   Let CU *be the set of identities queried to* $\mathcal{O}^{\mathsf{Corrupt}}$*.*

3. $\mathcal{A}$ *outputs a message* M^* *and a signature* Σ^**.*

$\mathcal{A}$ *succeeds if* $(i)\mathsf{Verify}(\mathsf{gpk}, M^*, \Sigma^*) = 1$ *and* $(ii)\mathsf{Sign}(\mathsf{gsk}[j], M^*)$ *was never queried for* $j \notin CU$*, and yet* $(iii)\mathsf{Open}(\mathsf{gmsk}, M^*, \Sigma^*) \notin CU$*.*

4.2.4 Transform of Index

Let $N-1 = q^\ell - 1$ be the number of users, $\mathcal{B} = \{\alpha_1, \ldots, \alpha_m\}$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, $p(X)$ be an irreducible polynomial of degree ℓ over $\mathbb{F}_q$, and $\mathbf{B} \in \mathbb{F}_q^{\ell \times mk}$ be a generator matrix of the systematic form of some q-ary linear code $\mathcal{C}$. We define a map $\mathsf{I2V}\colon [N-1] \longrightarrow \mathbb{F}_{q^m}^k$ as follows.

1. $f_1\colon [N-1] \longrightarrow \mathbb{F}_q^\ell$ is any public injective map such that $f_1(j) \neq \mathbf{0}$. For example, let α be a primitive element of $\mathbb{F}_q$, *i.e.*, $\mathbb{F}_q = \{0, \alpha, \ldots, \alpha^{q-1}\}$, and $f\colon [q-1] \cup \{0\} \to \mathbb{F}_q$ the map such that $f(0) = 0$ and $f(i) = \alpha^i$ for $i = 1, \ldots, q-1$, then $f_1(j) = \big(f(x_0), \ldots, f(x_{\ell-1})\big)$, where $j = x_0 + \cdots + x_{\ell-1}q^{\ell-1}$ is the representation of j in the base q.

2. $f_2\colon \mathbb{F}_q^\ell \longrightarrow \mathbb{F}_{q^m}^k$ defined as follows: for a vector $(a_0, \ldots, a_{\ell-1}) \in \mathbb{F}_q^\ell$, compute

$$(b_0, \ldots, b_{mk-1}) = (a_0, \ldots, a_{\ell-1}) \cdot \mathbf{B}$$

   and form the matrix

$$\mathbf{A} = \begin{pmatrix} b_0 & \cdots & b_{m-1} \\ \vdots & \vdots & \vdots \\ b_{(k-1)m} & \cdots & b_{mk-1} \end{pmatrix}.$$

   Then

$$f_2(a_0, \ldots, a_{\ell-1}) := (\alpha_1, \ldots, \alpha_m) \cdot \mathbf{A}^T.$$

3. Define $\mathsf{I2V}(j) := f_2 \circ f_1(j)$, where $\circ$ denotes the composition of mapping.

Let S denote the image of $\mathsf{I2V}$, then S is a subset of cardinality N of $\mathbb{F}_{q^m}^k$. Conversely, for each vector $\mathbf{m} = (m_1, \ldots, m_k) \in S$, there is a unique $j \in [N-1] \cup \{0\}$ such that $\mathsf{I2V}(j) = \mathbf{m}$. (If $\mathbf{m} = \mathbf{0}$, then j is set to be equal to 0.) The inverse map is denoted by $\mathsf{V2I} := f_1^{-1} \circ f_2^{-1}$.

4.2.5 Permutations

Let $\mathbf{v} \in \mathbb{F}_q^\ell \setminus \{\mathbf{0}\}$ be a random vector. We define two permutations:

- $P_\mathbf{v}\colon \mathbb{F}_q^{N-1} \longrightarrow \mathbb{F}_q^{N-1}$ transforms $\mathbf{x} = (x_1, \ldots, x_{N-1})$ to $\mathbf{x}' = (x'_1, \ldots, x'_{N-1})$, where $x_i = x'_{f_1^{-1}\left(f_1(i)\cdot\mathbf{v}\right)}$. Here, the multiplication is defined with respect to $p(X)$. Therefore,

$$\mathbf{x} = \delta_j \Longleftrightarrow P_\mathbf{v}(\mathbf{x}) = \delta^{N-1}_{f_1^{-1}\left(f_1(j)\cdot\mathbf{v}\right)}.$$

- $P'_{\mathbf{v}}\colon S \longrightarrow S$ as follows. For a vector $\mathbf{z} \in S$, let $\mathbf{z}_1 = f_2^{-1}(\mathbf{z}) \in \mathbb{F}_q^{\ell}$. Let $\mathbf{z}_2 = \mathbf{v} \cdot \mathbf{z}_1$ and define $P'_{\mathbf{v}}(\mathbf{z}) = f_2(\mathbf{z}_2)$. Since $f_2^{-1}\big(\mathsf{I2V}(j)\big) = f_1(j)$, so clearly,

$$\mathbf{m} = \mathsf{I2V}(j) \Longleftrightarrow P'_{\mathbf{v}}(\mathbf{m}) = f_2\big(f_1(j) \cdot \mathbf{v}\big).$$

Our construction also makes use of the operation "$\star$" as defined in Section 2.5.4.

4.3 The Underlying Interactive Protocol

4.3.1 The Interactive Scheme

This section is devoted to our zero-knowledge argument of knowledge. Let k, ℓ, m, $m_0, n, n_0, r_0, w_r, w_s$ be positive integers. The number of group users is $N-1 = q^{\ell}-1$. The common input contains matrices $\mathbf{H} \in \mathbb{F}_{q^{m_0}}^{r_0 \times n_0}, \mathbf{H}_1, \ldots, \mathbf{H}_4$, $N-1$ syndromes $\mathbf{y}_1, \ldots, \mathbf{y}_{N-1} \in \mathbb{F}_{q^{m_0}}^{r_0}$, a ciphertext $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2) \in \mathbb{F}_{q^m}^{2n}$, a basis $\mathcal{B} = \{\alpha_1, \ldots, \alpha_m\}$ of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, a generator matrix $\mathbf{B} \in \mathbb{F}_q^{\ell \times mk}$ of a code $\mathcal{C}$, and an irreducible polynomial $p(X)$ of degree ℓ over $\mathbb{F}_q[X]$, the two maps f_1 and f_2 described as above. The output of the protocol is that prover $\mathcal{P}$ simultaneously convinces verifier $\mathcal{V}$ in zero-knowledge that $\mathcal{P}$ possesses a vector $\mathbf{s} \in \mathcal{S}_{w_s}^{n_0, m_0}$ corresponding to a certain syndrome $\mathbf{y}_j \in \{\mathbf{y}_1, \ldots, \mathbf{y}_{N-1}\}$ with hidden index j, and that $\mathbf{c}$ is a correct encryption of $\mathbf{m} = \mathsf{I2V}(j)$ using the RQC scheme described by $\mathbf{H}_1, \ldots, \mathbf{H}_4$. More precisely, the secret witness of $\mathcal{P}$ is a tuple $(j, \mathbf{s}, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3) \in [N-1] \times \mathbb{F}_{q^{m_0}}^{n_0} \times \mathbb{F}_{q^m}^{n} \times \mathbb{F}_{q^m}^{n} \times \mathbb{F}_{q^m}^{n}$ such that

$$\begin{cases} \mathbf{H} \cdot \mathbf{s}^T = \mathbf{y}^T \quad \wedge \quad \mathbf{s} \in \mathcal{S}_{w_s}^{n_0, m_0}, \\ \widetilde{\mathbf{H}} \cdot \big(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathsf{I2V}(j)\big)^T = \mathbf{c}^T \quad \wedge \quad \mathbf{r}_i \in \mathcal{S}_{w_r}^{n,m}, i = 1, 2, 3, \end{cases}$$

where $\widetilde{\mathbf{H}} = [\mathbf{H}_1 | \cdots | \mathbf{H}_4]$. Let $\mathbf{A} = \big[\mathbf{y}_1^T | \cdots | \mathbf{y}_{N-1}^T\big] \in \mathbb{F}_{q^{m_0}}^{r_0 \times (N-1)}$, $\mathbf{m} = \mathsf{I2V}(j)$, and $\mathbf{x} = \delta_j^{N-1}$ be the index representation vector of j. Then, the above equations can be expressed as

$$\begin{cases} \mathbf{H} \cdot \mathbf{s}^T - \mathbf{A} \cdot \mathbf{x}^T = \mathbf{0} \quad \wedge \quad \mathbf{x} = \delta_j^{N-1} \quad \wedge \quad \mathbf{s} \in \mathcal{S}_{w_s}^{n_0, m_0}, \\ \widetilde{\mathbf{H}} \cdot \big(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{m}\big)^T = \mathbf{c}^T \quad \wedge \quad \mathbf{m} = \mathsf{I2V}(j) \quad \wedge \quad \mathbf{r}_i \in \mathcal{S}_{w_r}^{n,m}, i = 1, 2, 3. \end{cases}$$

A ZKAoK for the above relations is obtained as follows:

- To prove that $\mathbf{x} = \delta_j^{N-1}$ and $\mathbf{m} = \mathsf{I2V}(j)$ without revealing j, prover $\mathcal{P}$ randomly picks a vector $\mathbf{v} \in \mathbb{F}_q^{\ell} \setminus \{\mathbf{0}\}$, sends $\mathbf{v}' = f_1(j) \cdot \mathbf{v}$ and shows that

$$P_{\mathbf{v}}(\mathbf{x}) = \delta_{j'}^{N-1} \quad \text{and} \quad P'_{\mathbf{v}}(\mathbf{m}) = \mathsf{I2V}(j'),$$

  where $j' = f_1^{-1}(\mathbf{v}')$.

- To prove in zero-knowledge that $\mathbf{s} \in \mathcal{S}_{w_s}^{n_0,m_0}$, prover $\mathcal{P}$ chooses random matrices $\mathbf{Q}_0 \leftarrow \mathrm{GL}(m_0, q)$, $\mathbf{P}_0 \leftarrow \mathrm{GL}(n_0, q)$, and shows that $\mathbf{Q}_0 \star \mathbf{s}\mathbf{P}_0 \in \mathcal{S}_{w_s}^{n_0,m_0}$. To prove in zero-knowledge that $\mathbf{r}_i \in \mathcal{S}_{w_r}^{n,m}$ and that they share the same support, $\mathcal{P}$ samples randomly $\mathbf{Q} \leftarrow \mathrm{GL}(m, q), \mathbf{P}_i \leftarrow \mathrm{GL}(n, q)$ and shows that $\mathbf{Q} \star \mathbf{r}_i\mathbf{P}_i \in \mathcal{S}_{w_r}^{n,m}$ having the same support, for $i = 1, 2, 3$. We refer the reader to [BBB+21] for more details.
- To prove the linear equations in ZK, $\mathcal{P}$ samples $(\mathbf{v_s}, \mathbf{v_x}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v_m})$ randomly and shows that

$$\begin{cases} \mathbf{H} \cdot (\mathbf{s} + \mathbf{v_s})^T - \mathbf{A} \cdot (\mathbf{x} + \mathbf{v_x})^T = \mathbf{H} \cdot \mathbf{v_s}^T - \mathbf{A} \cdot \mathbf{v_x}^T, \\ \widetilde{\mathbf{H}} \cdot \big(\mathbf{r}_1 + \mathbf{v}_1, \mathbf{r}_2 + \mathbf{v}_2, \mathbf{r}_3 + \mathbf{v}_3, \mathbf{m} + \mathbf{v_m}\big)^T - \mathbf{c}^T = \widetilde{\mathbf{H}} \cdot \big(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v_m}\big)^T. \end{cases}$$

Finally, let $h\colon \{0,1\}^* \to \{0,1\}^\lambda$ be a collision-resistant hash function, the protocol is described as follows.

1. $\mathcal{P}$ samples

$$\begin{cases} \mathbf{Q}_0 \leftarrow \mathrm{GL}(m_0, q), \mathbf{Q} \leftarrow \mathrm{GL}(m, q); \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \leftarrow \mathrm{GL}(n, q); \mathbf{P}_0 \leftarrow \mathrm{GL}(n_0, q); \\ \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \leftarrow \mathbb{F}_{q^m}^n, \mathbf{v_m} \leftarrow S; \mathbf{v_x} \leftarrow \mathbb{F}_q^{N-1}, \mathbf{v_s} \leftarrow \mathbb{F}_{q^{m_0}}^{n_0}, \mathbf{v} \leftarrow \mathbb{F}_q^\ell; \\ \rho_1, \rho_2, \rho_3 \leftarrow 1^\lambda, \end{cases}$$

   and sends the commitment $\mathsf{CMT} = (c_1, c_2, c_3)$, where

$$\begin{cases} c_1 = h\big(\mathbf{v}, \mathbf{Q}, \mathbf{Q}_0, \mathbf{P}_0, \ldots, \mathbf{P}_3, \mathbf{H} \cdot \mathbf{v_s}^T - \mathbf{A} \cdot \mathbf{v_x}^T, \widetilde{\mathbf{H}} \cdot \big(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v_m}\big)^T, \rho_1\big), \\ c_2 = h\big(\mathbf{Q}_0 \star \mathbf{v_s}\mathbf{P}_0, P_{\mathbf{v}}(\mathbf{v_x}), P'_{\mathbf{v}}(\mathbf{v_m}), (\mathbf{Q} \star \mathbf{v}_i\mathbf{P}_i)_{i=1}^3, \rho_2\big), \\ c_3 = h\big(\mathbf{Q}_0 \star (\mathbf{s} + \mathbf{v_s})\mathbf{P}_0, P_{\mathbf{v}}(\mathbf{x} + \mathbf{v_x}), P'_{\mathbf{v}}(\mathbf{m} + \mathbf{v_m}), (\mathbf{Q} \star (\mathbf{r}_i + \mathbf{v}_i)\mathbf{P}_i)_{i=1}^3, \rho_3\big). \end{cases}$$

2. $\mathcal{V}$ sends a random challenge $\mathsf{Ch} \in \{1, 2, 3\}$ to $\mathcal{P}$.
3. $\mathcal{P}$ replies as:
   - If $\mathsf{Ch} = 1$, reveal c_2 and c_3. Let $\mathbf{v}' = f_1(j) \cdot \mathbf{v}$.

$$\begin{cases} \widehat{\mathbf{v}}_{\mathbf{s}} = \mathbf{Q}_0 \star \mathbf{v_s}\mathbf{P}_0, \\ \widehat{\mathbf{s}} = \mathbf{Q}_0 \star \mathbf{s}\mathbf{P}_0 \end{cases} \quad \widehat{\mathbf{v}}_{\mathbf{x}} = P_{\mathbf{v}}(\mathbf{v_x}), \quad \widehat{\mathbf{v}}_{\mathbf{m}} = P'_{\mathbf{v}}(\mathbf{v_m}), \quad \begin{cases} \widehat{\mathbf{v}}_i = \mathbf{Q} \star \mathbf{v}_i\mathbf{P}_i, \\ \widehat{\mathbf{r}}_i = \mathbf{Q} \star \mathbf{r}_i\mathbf{P}_i, \end{cases}$$

     $\mathcal{P}$ sends $\mathsf{RSP} = \big(\mathbf{v}', \widehat{\mathbf{s}}, \widehat{\mathbf{v}}_{\mathbf{s}}, \widehat{\mathbf{v}}_{\mathbf{x}}, \widehat{\mathbf{v}}_{\mathbf{m}}, (\widehat{\mathbf{v}}_i)_{i=1}^3, (\widehat{\mathbf{r}}_i)_{i=1}^3, \rho_2, \rho_3\big)$ to $\mathcal{V}$.
   - If $\mathsf{Ch} = 2$, reveal c_1 and c_3. Let

$$\begin{cases} \mathbf{v}'' = \mathbf{v}, \mathbf{E} = \mathbf{Q}, \mathbf{E}_0 = \mathbf{Q}_0, \mathbf{F}_i = \mathbf{P}_i, 0 \le i \le 3, \\ \mathbf{z_s} = \mathbf{s} + \mathbf{v_s}, \mathbf{z_x} = \mathbf{x} + \mathbf{v_x}, \mathbf{z_m} = \mathbf{m} + \mathbf{v_m}, \mathbf{z}_i = \mathbf{r}_i + \mathbf{v}_i, 1 \le i \le 3. \end{cases}$$

     $\mathcal{P}$ sends $\mathsf{RSP} = \big(\mathbf{v}'', \mathbf{E}, \mathbf{E}_0, (\mathbf{F}_i)_{i=0}^3, \mathbf{z_s}, \mathbf{z_x}, \mathbf{z_m}, (\mathbf{z}_i)_{i=1}^3, \rho_1, \rho_3\big)$ to $\mathcal{V}$.

– If $\mathsf{Ch} = 3$, reveal c_1 and c_2. Let

$$\begin{cases} \mathbf{v}''' = \mathbf{v}, \mathbf{U} = \mathbf{Q}, \mathbf{U}_0 = \mathbf{Q}_0, \mathbf{V}_i = \mathbf{P}_i, 0 \leq i \leq 3, \\ \mathbf{y_s} = \mathbf{v_s}, \mathbf{y_x} = \mathbf{v_x}, \mathbf{y_m} = \mathbf{v_m}, \mathbf{y}_i = \mathbf{v}_i, 1 \leq i \leq 3. \end{cases}$$

$\mathcal{P}$ sends $\mathsf{RSP} = \big(\mathbf{v}''', \mathbf{U}, \mathbf{U}_0, (\mathbf{V}_i)_{i=0}^3, \mathbf{y_s}, \mathbf{y_x}, \mathbf{y_m}, (\mathbf{y}_i)_{i=1}^3, \rho_1, \rho_2\big)$ to $\mathcal{V}$.

4. $\mathcal{V}$ performs the following checks:

– If $\mathsf{Ch} = 1$, let $\mathbf{w_x} = \delta_{f_1^{-1}(\mathbf{v}')} \in \mathbb{F}_q^{N-1}$ and $\mathbf{w_m} = f_2(\mathbf{v}') \in \mathbb{F}_{q^m}^k$. Check that $\widehat{\mathbf{s}} \in \mathcal{S}_{w_s}^{n_0,m_0}$ and $\widehat{\mathbf{r}}_i \in \mathcal{S}_{w_r}^{n,m}$ have the same support, and that

$$\begin{cases} c_2 = h\big(\widehat{\mathbf{s}}, \widehat{\mathbf{v}}_\mathbf{x}, \widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, \widehat{\mathbf{v}}_3, \widehat{\mathbf{v}}_\mathbf{m}, \rho_2\big), \\ c_3 = h\big(\widehat{\mathbf{s}} + \widehat{\mathbf{v}}_\mathbf{s}, \widehat{\mathbf{v}}_\mathbf{x} + \mathbf{w_x}, \widehat{\mathbf{v}}_\mathbf{m} + \mathbf{w_m}, (\widehat{\mathbf{r}}_i + \widehat{\mathbf{v}}_i)_{i=1}^3, \rho_3\big). \end{cases}$$

– If $\mathsf{Ch} = 2$, check that

$$\begin{cases} c_1 = h\left(\mathbf{v}'', \mathbf{E}, \mathbf{E}_0, \mathbf{F}_0, \ldots, \mathbf{F}_3, \mathbf{H} \cdot \mathbf{z}_\mathbf{s}^T - \mathbf{A} \cdot \mathbf{z}_\mathbf{x}^T, \widetilde{\mathbf{H}} \cdot \big(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z_m}\big)^T - \mathbf{c}^T, \rho_1\right), \\ c_3 = h\big(\mathbf{E}_0 \star \mathbf{z_s}\mathbf{F}_0, P_{\mathbf{v}''}(\mathbf{z_x}), P'_{\mathbf{v}''}(\mathbf{z_m}), (\mathbf{E} \star \mathbf{z}_i\mathbf{F}_i)_{i=1}^3, \rho_3\big). \end{cases}$$

– If $\mathsf{Ch} = 3$, check that

$$\begin{cases} c_1 = h\left(\mathbf{v}''', \mathbf{U}, \mathbf{U}_0, \mathbf{V}_0, \ldots, \mathbf{V}_3, \mathbf{H} \cdot \mathbf{y}_\mathbf{s}^T - \mathbf{A} \cdot \mathbf{y}_\mathbf{x}^T, \widetilde{\mathbf{H}} \cdot \big(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y_m}\big)^T, \rho_1\right), \\ c_2 = h\big(\mathbf{U}_0 \star \mathbf{y_s}\mathbf{V}_0, P_{\mathbf{v}'''}(\mathbf{y_x}), P'_{\mathbf{v}'''}(\mathbf{y_m}), (\mathbf{U} \star \mathbf{y}_i\mathbf{V}_i)_{i=1}^3, \rho_2\big). \end{cases}$$

5. $\mathcal{V}$ outputs 1 it all checks are passed; otherwise, it outputs 0.

4.3.2 Analysis

Proposition 4.1. *The above interactive protocol has perfect completeness, and has communication cost bounded by* $C = (\ell + N - 1 + m_0^2 + m_0 n_0 + n_0^2 + m^2 + 3mn + 3n^2 + km) \log q + 5\lambda$. *It is a statistical zero-knowledge argument in the random oracle model.*

Communication Cost:

- The commitment CMT has bit-size 3λ.
- For $\mathsf{Ch} = 1$, we have $C_1 = (\ell + 2n_0 m_0 + N - 1 + km + 6nm) \log q + 2\lambda$.
- For $\mathsf{Ch} = 2$ or 3, we have

$$C_{2,3} = (\ell + N - 1 + m_0^2 + m_0 n_0 + n_0^2 + m^2 + 3mn + 3n^2 + km) \log q + 2\lambda.$$

The total cost is bounded by

$$C = (\ell + N - 1 + m_0^2 + m_0 n_0 + n_0^2 + m^2 + 3mn + 3n^2 + km)\log q + 5\lambda.$$

Zero-knowledge Property.

Lemma 4. *In the random oracle model, there exists an efficient simulator $\mathcal{S}$ interacting with a verifier $\widehat{\mathcal{V}}$, such that, given only the public input of the protocol, $\mathcal{S}$ outputs with probability negligibly close to $\frac{2}{3}$ a simulated transcript that is statistically close to the one produced by the honest prover in the real interaction.*

Proof. Simulator $\mathcal{S}$, given the public input $(\mathbf{H}, \mathbf{H}_1, \ldots, \mathbf{H}_4, \mathbf{A}, \mathbf{c})$, starts by picking a random $\overline{\mathsf{Ch}} \in \{1, 2, 3\}$. Next, we consider 3 cases.

Case 1: $\overline{\mathsf{Ch}} = 1$, $\mathcal{S}$ proceeds as follows:

1. Compute $\mathbf{s}' \in \mathbb{F}_{q^{m_0}}^{n_0}$ and $\mathbf{x}' \in \mathbb{F}_q^{N-1}$ satisfying $\mathbf{H} \cdot \mathbf{s}'^T = \mathbf{A} \cdot \mathbf{x}'^T$, and $\mathbf{m}' \in S, \mathbf{r}_1', \mathbf{r}_2', \mathbf{r}_3' \in \mathbb{F}_{q^m}^n$ such that $\widetilde{\mathbf{H}} \cdot \big(\mathbf{r}_1', \mathbf{r}_2', \mathbf{r}_3', \mathbf{m}'\big)^T = \mathbf{c}^T$.

2. Sample random objects, compute and send a commitment as in the real scheme. Namely, $\mathcal{S}$ samples

$$\begin{cases} \mathbf{Q}_0 \leftarrow \mathrm{GL}(m_0, q), \mathbf{Q} \leftarrow \mathrm{GL}(m, q); \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \leftarrow \mathrm{GL}(n, q); \mathbf{P}_0 \leftarrow \mathrm{GL}(n_0, q); \\ \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \leftarrow \mathbb{F}_{q^m}^n, \mathbf{v}_\mathbf{m} \leftarrow S; \mathbf{v}_\mathbf{x} \leftarrow \mathbb{F}_q^{N-1}, \mathbf{v}_\mathbf{s} \leftarrow \mathbb{F}_{q^{m_0}}^{n_0}, \mathbf{v} \leftarrow \mathbb{F}_q^\ell; \\ \rho_1, \rho_2, \rho_3 \leftarrow 1^\lambda, \end{cases}$$

   and sends the commitment $\mathsf{CMT} = (c_1', c_2', c_3')$, where

$$\begin{cases} c_1 = h\big(\mathbf{v}, \mathbf{Q}, \mathbf{Q}_0, \mathbf{P}_0, \ldots, \mathbf{P}_3, \mathbf{H} \cdot \mathbf{v}_\mathbf{s}^T - \mathbf{A} \cdot \mathbf{v}_\mathbf{x}^T, \widetilde{\mathbf{H}} \cdot \big(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_\mathbf{m}\big)^T, \rho_1\big), \\ c_2 = h\big(\mathbf{Q}_0 \star \mathbf{v}_\mathbf{s}\mathbf{P}_0, P_\mathbf{v}(\mathbf{v}_\mathbf{x}), P'_\mathbf{v}(\mathbf{v}_\mathbf{m}), (\mathbf{Q} \star \mathbf{v}_i\mathbf{P}_i)_{i=1}^3, \rho_2\big), \\ c_3 = h\big(\mathbf{Q}_0 \star (\mathbf{s} + \mathbf{v}_\mathbf{s})\mathbf{P}_0, P_\mathbf{v}(\mathbf{x} + \mathbf{v}_\mathbf{x}), P'_\mathbf{v}(\mathbf{m} + \mathbf{v}_\mathbf{m}), (\mathbf{Q} \star (\mathbf{r}_i + \mathbf{v}_i)\mathbf{P}_i)_{i=1}^3, \rho_3\big). \end{cases}$$

Receiving a challenge Ch from $\widehat{\mathcal{V}}$, the simulator responds as follows:

- If $\mathsf{Ch} = 1$: Output $\perp$, and abort.
- If $\mathsf{Ch} = 2$: Send

$$\mathsf{RSP} = \big(\mathbf{v}, \mathbf{Q}, \mathbf{Q}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{s}' + \mathbf{v}_\mathbf{s}, \mathbf{x}' + \mathbf{v}_\mathbf{x}, \mathbf{m}' + \mathbf{v}_\mathbf{m}, (\mathbf{r}_i' + \mathbf{v}_i)_{i=1}^3; \rho_1, \rho_3\big).$$

- If $\mathsf{Ch} = 3$: Send

$$\mathsf{RSP} = \big(\mathbf{v}, \mathbf{Q}, \mathbf{Q}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \mathbf{v}_\mathbf{s}, \mathbf{v}_\mathbf{x}, \mathbf{v}_\mathbf{m}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3; \rho_1, \rho_2\big).$$

Case 2: $\overline{\mathsf{Ch}} = 2$, $\mathcal{S}$ samples

$$\begin{cases} j' \leftarrow [N-1], \mathbf{s}' \leftarrow \mathcal{S}_{w_s}^{n_0,m_0}, \mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3 \in \mathcal{S}_{w_r}^{n,m}; \\ \mathbf{Q}_0 \leftarrow \mathrm{GL}(m_0, q), \mathbf{Q} \leftarrow \mathrm{GL}(m, q); \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3 \leftarrow \mathrm{GL}(n, q); \mathbf{P}_0 \leftarrow \mathrm{GL}(n_0, q); \\ \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \leftarrow \mathbb{F}_{q^m}^n, \mathbf{v_m} \leftarrow S; \mathbf{v_x} \leftarrow \mathbb{F}_q^{N-1}, \mathbf{v_s} \leftarrow \mathbb{F}_{q^{m_0}}^{n_0}, \mathbf{v} \leftarrow \mathbb{F}_q^{\ell}; \\ \rho_1, \rho_2, \rho_3 \leftarrow 1^{\lambda}, \end{cases}$$

and let $\mathbf{x}' = \delta_{j'}^{N-1}$, and $\mathbf{m}' = \mathsf{I2V}(j')$. $\mathcal{S}$ sends the commitment computed as in the case $\overline{\mathsf{Ch}} = 1$. After receiving a challenge Ch from $\widehat{\mathcal{V}}$, it responds as follows:

- If $\mathsf{Ch} =: 1$ Send

  $$\mathsf{RSP} = \big(\mathsf{I2V}(j') + \mathbf{v}, \mathbf{Q}_0 \star \mathbf{s}\mathbf{P}_0, \mathbf{Q}_0 \star \mathbf{v_s}\mathbf{P}_0, P_{\mathbf{v}}(\mathbf{v_x}), P'_{\mathbf{v}}(\mathbf{v_m}), \mathbf{Q} \star \mathbf{v}_i\mathbf{P}_i, \mathbf{Q} \star \mathbf{r}'_i\mathbf{P}_i; \rho_2, \rho_3\big),$$

  which contains all $i = 1, 2, 3$.

- If $\mathsf{Ch} = 2$: Output $\perp$, and abort.

- If $\mathsf{Ch} = 3$: Send RSP computed as in the case $(\overline{\mathsf{Ch}} = 1, \mathsf{Ch} = 3)$.

Case 3: $\overline{\mathsf{Ch}} = 3$, the simulator performs the preparation as in the case $\overline{\mathsf{Ch}} = 2$. It sends the commitment $\mathsf{CMT} = (c'_1, c'_2, c'_3)$, where c'_2 and c'_3 are computed as usual, while

$$c'_1 = h\left(\mathbf{v}, \mathbf{Q}, \mathbf{Q}_0, \mathbf{P}_0, \ldots, \mathbf{P}_3, \mathbf{H} \cdot (\mathbf{s}' + \mathbf{v_s})^T - \mathbf{A} \cdot (\mathbf{x}' + \mathbf{v_x})^T, \widetilde{\mathbf{H}} \cdot \mathbf{z}^T, \rho_1\right),$$

where $\mathbf{z} = \big(\mathbf{r}'_1 + \mathbf{v}_1, \mathbf{r}'_2 + \mathbf{v}_2, \mathbf{r}'_3 + \mathbf{v}_3, \mathbf{m}' + \mathbf{v_m}\big)$. Next, after receiving a challenge Ch, $\mathcal{S}$ responds as follows:

- If $\mathsf{Ch} = 1$: Send RSP as in the case $(\overline{\mathsf{Ch}} = 2, \mathsf{Ch} = 1)$.

- If $\mathsf{Ch} = 2$: Send RSP as in the case $(\overline{\mathsf{Ch}} = 1, \mathsf{Ch} = 2)$.

- If $\mathsf{Ch} = 3$: Output $\perp$, and abort.

Since the challenge is a random value from $\{1, 2, 3\}$, so the probability that $\mathcal{S}$ outputs $\perp$ is $\frac{1}{3}$. In the cases that $\mathcal{S}$ does not output $\perp$, one can easily verify that the distribution of its output is identical to that in the real interaction. □ □

Soundness Property.

Lemma 5. *Given the public input of the protocol, a commitment* CMT *and* 3 *valid responses* $\mathsf{RSP}_1, \mathsf{RSP}_2, \mathsf{RSP}_3$ *to all possible values of the challenge* Ch*, one can efficiently construct a knowledge extractor* $\mathcal{E}$ *that outputs a tuple*

$$(j', \mathbf{s}', \mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3) \in [N-1] \times \mathbb{F}_{q^{m_0}}^{n_0} \times \mathbb{F}_{q^m}^{n} \times \mathbb{F}_{q^m}^{n} \times \mathbb{F}_{q^m}^{n}$$

such that

$$\begin{cases} \mathbf{H} \cdot \mathbf{s}'^T = \mathbf{y}'^T \quad \textit{and} \quad \mathbf{s}' \in \mathcal{S}_{w_s}^{n_0,m_0}, \\ \widetilde{\mathbf{H}} \cdot \big(\mathbf{r}'_1, \mathbf{r}'_2, \mathbf{r}'_3, \mathsf{I2V}(j')\big)^T = \mathbf{c}^T \quad \textit{and} \quad \mathbf{r}'_i \in \mathcal{S}_{w_r}^{n,m}, i = 1, 2, 3. \end{cases}$$

Proof. Assume that we have a commitment $\mathsf{CMT} = (c_1, c_2, c_3)$ and 3 responses

$$\begin{cases} \mathsf{RSP}_1 = \big(\mathbf{v}', \widehat{\mathbf{s}}, \widehat{\mathbf{v}}_{\mathbf{s}}, \widehat{\mathbf{v}}_{\mathbf{x}}, \widehat{\mathbf{v}}_{\mathbf{m}}, (\widehat{\mathbf{v}}_i)_{i=1}^3, (\widehat{\mathbf{r}}_i)_{i=1}^3, \rho_2, \rho_3\big), \\ \mathsf{RSP}_2 = \big(\mathbf{v}'', \mathbf{E}, \mathbf{E}_0, (\mathbf{F}_i)_{i=0}^3, \mathbf{z}_{\mathbf{s}}, \mathbf{z}_{\mathbf{x}}, \mathbf{z}_{\mathbf{m}}, (\mathbf{z}_i)_{i=1}^3, \rho_1, \rho_3\big), \\ \mathsf{RSP}_3 = \big(\mathbf{v}''', \mathbf{U}, \mathbf{U}_0, (\mathbf{V}_i)_{i=0}^3, \mathbf{y}_{\mathbf{s}}, \mathbf{y}_{\mathbf{x}}, \mathbf{y}_{\mathbf{m}}, (\mathbf{y}_i)_{i=1}^3, \rho_1, \rho_2\big) \end{cases}$$

that satisfy all the verification conditions with respect to $\mathsf{Ch} = 1, 2, 3$, respectively. Thus, we have the following relations:

$$\begin{cases} \widehat{\mathbf{s}} \in \mathcal{S}_{w_s}^{n_0,m_0}, \mathbf{w}_{\mathbf{x}} = \delta_{f_1^{-1}(\mathbf{v}')}^{N-1}, \mathbf{w}_{\mathbf{m}} = f_2(\mathbf{v}'), \widehat{\mathbf{r}}_i \in \mathcal{S}_{w_r}^{n,m}, \\ c_1 = h\left(\mathbf{v}'', \mathbf{E}, \mathbf{E}_0, \mathbf{F}_0, \ldots, \mathbf{F}_3, \mathbf{H} \cdot \mathbf{z}_{\mathbf{s}}^T - \mathbf{A} \cdot \mathbf{z}_{\mathbf{x}}^T, \widetilde{\mathbf{H}} \cdot \big(\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_{\mathbf{m}}\big)^T - \mathbf{c}^T, \rho_1\right), \\ c_1 = h\left(\mathbf{v}''', \mathbf{U}, \mathbf{U}_0, \mathbf{V}_0, \ldots, \mathbf{V}_3, \mathbf{H} \cdot \mathbf{y}_{\mathbf{s}}^T - \mathbf{A} \cdot \mathbf{y}_{\mathbf{x}}^T, \widetilde{\mathbf{H}} \cdot \big(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_{\mathbf{m}}\big)^T, \rho_1\right), \\ c_2 = h\big(\widehat{\mathbf{s}}, \widehat{\mathbf{v}}_{\mathbf{x}}, \widehat{\mathbf{v}}_1, \widehat{\mathbf{v}}_2, \widehat{\mathbf{v}}_3, \widehat{\mathbf{v}}_{\mathbf{m}}, \rho_2\big), \\ c_2 = h\big(\mathbf{U}_0 \star \mathbf{y}_{\mathbf{s}}\mathbf{V}_0, P_{\mathbf{v}'''}(\mathbf{y}_{\mathbf{x}}), P'_{\mathbf{v}'''}(\mathbf{y}_{\mathbf{m}}), (\mathbf{U} \star \mathbf{y}_i\mathbf{V}_i)_{i=1}^3, \rho_2\big), \\ c_3 = h\big(\widehat{\mathbf{s}} + \widehat{\mathbf{v}}_{\mathbf{s}}, \widehat{\mathbf{v}}_{\mathbf{x}} + \mathbf{w}_{\mathbf{x}}, \widehat{\mathbf{v}}_{\mathbf{m}} + \mathbf{w}_{\mathbf{m}}, (\widehat{\mathbf{r}}_i + \widehat{\mathbf{v}}_i)_{i=1}^3, \rho_3\big), \\ c_3 = h\big(\mathbf{E}_0 \star \mathbf{z}_{\mathbf{s}}\mathbf{F}_0, P_{\mathbf{v}''}(\mathbf{z}_{\mathbf{x}}), P'_{\mathbf{v}''}(\mathbf{z}_{\mathbf{m}}), (\mathbf{E} \star \mathbf{z}_i\mathbf{F}_i)_{i=1}^4, \rho_3\big). \end{cases}$$

Since h is a collision-resistant hash function, it must be that:

$$\begin{cases} \mathbf{v}'' = \mathbf{v}''', \mathbf{E} = \mathbf{U}, \mathbf{E}_0 = \mathbf{U}_0, \mathbf{F}_i = \mathbf{V}_i, \\ \delta_{f_1^{-1}(\mathbf{v}')}^{N-1} = \mathbf{w}_{\mathbf{x}} = P_{\mathbf{v}''}(\mathbf{z}_{\mathbf{x}}) - P_{\mathbf{v}'''}(\mathbf{y}_{\mathbf{x}}) = P_{\mathbf{v}''}(\mathbf{z}_{\mathbf{x}} - \mathbf{y}_{\mathbf{x}}), \\ f_2(\mathbf{v}') = \mathbf{w}_{\mathbf{m}} = P'_{\mathbf{v}''}(\mathbf{z}_{\mathbf{m}}) - P'_{\mathbf{v}'''}(\mathbf{y}_{\mathbf{m}}) = P'_{\mathbf{v}''}(\mathbf{z}_{\mathbf{m}} - \mathbf{y}_{\mathbf{m}}), \\ \widehat{\mathbf{s}} = \mathbf{E}_0 \star \mathbf{z}_{\mathbf{s}}\mathbf{F}_0 - \mathbf{U}_0 \star \mathbf{y}_{\mathbf{s}}\mathbf{V}_0 = \mathbf{E}_0 \star (\mathbf{z}_{\mathbf{s}} - \mathbf{y}_{\mathbf{s}})\mathbf{F}_0 \in \mathcal{S}_{w_s}^{n_0,m_0}, \\ \widehat{\mathbf{r}}_i = \mathbf{E} \star \mathbf{z}_i\mathbf{F}_i - \mathbf{U} \star \mathbf{y}_i\mathbf{V}_i = \mathbf{E} \star (\mathbf{z}_i - \mathbf{y}_i)\mathbf{F}_i \in \mathcal{S}_{w_r}^{n,m}, \\ \mathbf{H} \cdot (\mathbf{z}_{\mathbf{s}} - \mathbf{y}_{\mathbf{s}})^T - \mathbf{A} \cdot (\mathbf{z}_{\mathbf{x}} - \mathbf{y}_{\mathbf{x}})^T = \mathbf{0}, \\ \widetilde{\mathbf{H}} \cdot \big(\mathbf{z}_1 - \mathbf{y}_1, \mathbf{z}_2 - \mathbf{y}_2, \mathbf{z}_3 - \mathbf{y}_3, \mathbf{z}_{\mathbf{m}} - \mathbf{y}_{\mathbf{m}}\big)^T = \mathbf{c}^T. \end{cases}$$

Now, let

- $j' = f_1^{-1}(\mathbf{z}_{\mathbf{x}} - \mathbf{y}_{\mathbf{x}}) \in [N-1]$,

- $\mathbf{s}' = \mathbf{z}_\mathbf{s} - \mathbf{y}_\mathbf{s} \in \mathcal{S}_{w_s}^{n_0,m_0}$,
- $\mathbf{r}'_i = \mathbf{z}_i - \mathbf{y}_i \in \mathcal{S}_{w_r}^{n,m}$,

It is easy to see that they satisfy the lemma. □ □

4.4 Our Code-Based Group Signature Scheme

1. **KeyGen**$(1^\lambda, N-1)$: On input a security parameter λ and an expected number of group users $N-1 = q^\ell - 1$, the algorithm first prepares as follows:
   - A primitive element α of $\mathbb{F}_q$ to describe the map f_1.
   - Parameters $m = m(\lambda), n = n(\lambda), k = k(\lambda), t = t(\lambda)$ for a rank GABIDULIN code $[n, k, 2t+1]$ over $\mathbb{F}_{q^m}$. In addition, it chooses an irreducible polynomial $F(X) \in \mathbb{F}_q[X]$ of degree m to define $\mathbb{F}_{q^m}$ as $\mathbb{F}_q[X]/\langle F\rangle$, and an irreducible polynomial $F_0(X) \in \mathbb{F}_q[X]$ of degree m_0 to define $\mathbb{F}_{q^{m_0}}$ as $\mathbb{F}_q[X]/\langle F_0\rangle$.
   - Parameters $w_r = w_r(\lambda)$ and an irreducible polynomial $P(X) \in \mathbb{F}_q[X]$ of degree n such that $P(X)$ is also irreducible over $\mathbb{F}_{q^m}$.
   - Parameters $n_0 = n_0(\lambda), r_0 = r_0(\lambda), w_s = w_s(\lambda)$ for the syndrome decoding problem. We choose $r_0 = \frac{1}{2}n_0$.
   - An irreducible polynomial $p(X)$ of degree ℓ over $\mathbb{F}_q$.
   - A generator matrix $\mathbf{M} \in \mathbb{F}_q^{\ell \times mk}$ of systematic form of a public linear code $\mathcal{C}$ over $\mathbb{F}_q$.
   - Two collision-resistant hash functions h and $\mathcal{H}$ used for generating commitments and random challenges, respectively.

   Then the algorithm performs the following steps:

   1. Run $\mathsf{RQC}(m, n, k, w_r, P(X))$ for a key pair $\big(\mathsf{pk}_\mathsf{RQC} = (\mathbf{H}_1, \ldots, \mathbf{H}_4), \mathsf{sk}_\mathsf{RQC}\big)$.
   2. The matrix $\mathbf{H}$ is constructed in the following way: Choose an irreducible polynomial $P_0(X) \in \mathbb{F}_q[X]$ of degree r_0 so that it is also irreducible over $\mathbb{F}_{q^{m_0}}$, a random vector $\mathbf{h}_0 \in \mathbb{F}_{q^{m_0}}^{r_0}$; then $\mathbf{H}$ is the ideal matrix generated by $\big(\mathbf{h}_0, P_0(X)\big)$.
   3. For each $j \in [N-1]$, choose $\mathbf{s}_j \leftarrow \mathcal{S}_{w_s}^{n_0,m_0}$ and let $\mathbf{y}_j^T = \mathbf{H} \cdot \mathbf{s}_j^T$, set $\mathbf{A} = \big[\mathbf{y}_1^T | \cdots | \mathbf{y}_{N-1}^T\big]$.
   4. Set $\widetilde{\mathbf{H}} = \big[\mathbf{H}_1 | \cdots | \mathbf{H}_4\big]$ and output

      $\mathsf{gpk} = (\mathbf{H}, \widetilde{\mathbf{H}}, \mathbf{A}, p, P, P_0, F, F_0, \mathbf{M}, \alpha), \mathsf{gmsk} = \mathsf{sk}_\mathsf{RQC}, \mathsf{gsk} = (\mathbf{s}_1, \ldots, \mathbf{s}_{N-1})$.

2. **Sign**$(\mathsf{gsk}[j], M)$: To sign a message $M \in \{0,1\}^*$ under gpk, the group user of index j performs the following steps:
   - Encrypt the representation vector of j, *i.e.,* the vector $\mathsf{I2V}(j) \in \mathbb{F}_{q^m}^k$, using $\mathsf{pk}_{\mathsf{RQC}}$.
   - Generate an $\mathsf{NIZKAoK}$ Π to simultaneously prove the possession of a vector $\mathbf{s} \in \mathcal{S}_{w_s}^{n_0,m_0}$ corresponding to a certain syndrome $\mathbf{y} \in \{\mathbf{y}_1, \ldots, \mathbf{y}_{N-1}\}$ with hidden index j, and that $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$ is a correct RQC encryption of $\mathsf{I2V}(j)$. This is done by using the interactive protocol in the above section with public input $(\mathbf{H}, \mathbf{H}_1, \ldots, \mathbf{H}_4, \mathbf{A}, \mathbf{c})$ and prover's witness $(j, \mathbf{s}, \mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3)$ which satisfy
   $$\begin{cases} \mathbf{H} \cdot \mathbf{s}^T = \mathbf{y}_j^T \quad \text{and} \quad \mathbf{s} \in \mathcal{S}_{w_s}^{n_0,m_0}, \\ \left[\mathbf{H}_1 | \cdots | \mathbf{H}_4\right] \cdot \left(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathsf{I2V}(j)\right)^T = \left(\mathbf{c}_1, \mathbf{c}_2\right)^T. \end{cases}$$
   The protocol is repeated $\kappa = \omega(\log \lambda)$ times to achieve negligible soundness error, and then made non-interactive, *i.e.,* we have
   $$\Pi = (\mathsf{CMT}^1, \ldots, \mathsf{CMT}^\kappa; (\mathsf{Ch}^1, \ldots, \mathsf{Ch}^\kappa); \mathsf{RSP}^1, \ldots, \mathsf{RSP}^\kappa),$$
   where$(\mathsf{Ch}^1, \ldots, \mathsf{Ch}^\kappa) = \mathcal{H}(M; \mathsf{CMT}^1, \ldots, \mathsf{CMT}^\kappa; \mathsf{gpk}, \mathbf{c})$.
   - Output the group signature $\Sigma = (\mathbf{c}, \Pi)$.
3. **Verify**$(\mathsf{gpk}, M, \Sigma)$: Parse Σ as $(\mathbf{c}, \Pi)$, and parse Π as above, and proceed as follows:
   - If $(\mathsf{Ch}^1, \ldots, \mathsf{Ch}^\kappa) \neq \mathcal{H}(M; \mathsf{CMT}^1, \ldots, \mathsf{CMT}^\kappa; \mathsf{gpk}, \mathbf{c})$, then return 0.
   - For $i = 1$ to κ, run the verification step of the interactive protocol with public input $(\mathbf{H}, \mathbf{H}_1, \ldots, \mathbf{H}_4, \mathbf{A}, \mathbf{c})$ to check the validity of RSP^i with respect to CMT^i and Ch^i. If any of the verification does not hold true, then return 0.
   - Return 1.
4. **Open**$(\mathsf{gmsk}, M, \Sigma)$: Parse Σ as $(\mathbf{c}, \Pi)$ and run $\mathsf{RQC.Dec}(\mathsf{gmsk}, \mathbf{c})$ to decrypt $\mathbf{c}$. If the decryption fails, then return $\perp$. If the decryption outputs $\mathbf{v} \in \mathbb{F}_{q^m}^k$, then return $j = \mathsf{V2I}(\mathbf{v}) \in [N-1]$.

The security of the scheme is stated in the following theorem.

Theorem 4.1. *In the random oracle model:*

- *If the decisional* $2-\mathsf{IRSD}(n, w_r)$ *and* $3-\mathsf{IRSD}(n, w_r)$ *problems are hard, then the scheme is* CPA- *anonymous.*
- *If the ideal rank syndrome decoding problem* $2-\mathsf{IRSD}(n_0, w_s)$ *is hard, then the scheme is traceable.*

4.4.1 Efficiency and Correctness

Efficiency. The size of gpk is dominated by $T \cdot \log q$, where

$$T = r_0 m_0 N + (k+2)nm + (\ell + n + r_0 + m_0 + m) + \ell km.$$

The length of the $\mathsf{NIZKAoK}$ is κ times the communication cost of the underlying interactive protocol. Therefore, the size of Σ is bounded by

$$\big((\ell + N - 1 + m_0^2 + m_0 n_0 + n_0^2 + m^2 + 3mn + 3n^2 + km)\log q + 5\lambda\big) \cdot \kappa + n.$$

Correctness. By guaranteeing that the user is honest and the underlying interactive protocol is perfectly complete, the correctness of the scheme is easily verified.

4.4.2 Anonymity

Let $\mathcal{A}$ be a PPT adversary attacking the CPA-anonymity of the scheme with advantage ε. We prove that ε is a negligible function of λ by considering the following sequence of experiments.

Experiment $G_0^{(b)}$. The challenger runs **KeyGen** to obtain

$$\begin{cases} \mathsf{gpk} = \big(\mathbf{H}, \widetilde{\mathbf{H}}, \mathbf{A}, p, P, P_0, F, F_0, \mathbf{M}, \alpha\big), \\ \mathsf{gmsk} = \mathsf{sk}_{\mathsf{RQC}}, \\ \mathsf{gsk} = (\mathbf{s}_1, \ldots, \mathbf{s}_{N-1}), \end{cases}$$

then gives gpk and gsk to $\mathcal{A}$. In the challenge phase, $\mathcal{A}$ outputs a message M^* and two indices $j_0, j_1 \in [N-1]$. The challenger sends back a challenge signature

$$\Sigma^* = (\mathbf{c}^*, \Pi^*) \leftarrow \mathbf{Sign}(\mathbf{s}_{j_b}, M^*).$$

The adversary outputs b with probability $\frac{1}{2} + \varepsilon$.

Experiment $G_1^{(b)}$. The challenge simulates Π^* as follows:

1. Compute $\mathbf{c}^*$ as in the Experiment $G_0^{(b)}$.
2. Run the simulator of the underlying interactive protocol and programming $\mathcal{H}$ accordingly.

3 Output the simulated of Π^*.

By the property of the simulator, we have $G_0^{(b)}$ and $G_1^{(b)}$ are statistically closed.

Experiment $G_2^{(b)}$. In this experiment, the vector $\mathbf{s}$ is replaced by a random vector in $\mathbb{F}_{q^m}^n$. By the hardness of the decisional 2-$\mathsf{IRSD}(n, w_r)$ problem, the adversary cannot distinguish a real public key $\mathbf{s}$ from a random one. Thus, Experiments $G_2^{(b)}$ and $G_1^{(b)}$ are computationally indistinguishable.

Experiment $G_3^{(b)}$. The vectors $\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3$ are replaced by random vectors $\overline{\mathbf{r}}_1, \overline{\mathbf{r}}_2, \overline{\mathbf{r}}_3 \leftarrow \mathbb{F}_{q^m}^n$. By the hardness of the decisional $3 - \mathsf{IRSD}(n, w_r)$ problem, we have $\mathbf{r}_1 + \mathbf{h} \cdot \mathbf{r}_2$ and $\mathbf{r}_3 + \mathbf{s} \cdot \mathbf{r}_2$ are computationally indistinguishable from $\overline{\mathbf{r}}_1 + \mathbf{h} \cdot \overline{\mathbf{r}}_2$ and $\overline{\mathbf{r}}_3 + \mathbf{s} \cdot \overline{\mathbf{r}}_2$, respectively. As a consequence, $\mathbf{r}_3 + \mathbf{s} \cdot \mathbf{r}_2 + \mathsf{I2V}(j_b) \cdot \mathbf{G}$ and $\overline{\mathbf{r}}_3 + \mathbf{s} \cdot \overline{\mathbf{r}}_2 + \mathsf{I2V}(j_b) \cdot \mathbf{G}$ are computationally indistinguishable. Therefore, Experiments $G_2^{(b)}$ and $G_3^{(b)}$ are computationally indistinguishable.

Experiment G_4. In this experiment, the ciphertext is set as $\mathbf{c}^* \leftarrow \mathbb{F}_{q^m}^{2n}$. It is evident that the distribution of $\mathbf{c}^*$ in Experiments $G_3^{(b)}$ and G_4 are identical, and hence, $G_3^{(b)}$ and G_4 are statistically indistinguishable. Observe that the ciphertext now no longer depends on the challenger's bit b, therefore, $\mathcal{A}$'s advantage in this experiment is 0.

The above arguments show that the advantage of $\mathcal{A}$ in $G_0^{(b)}$ is negligible, *i.e.,* ε is negligible. Thus, the scheme is CPA-anonymous.

4.4.3 Traceability

The proof of this property is quite similar to that of [ELL+15]. The only difference is that our proof is for rank metric. We include it here for the sake of completeness.

Assume that $\mathcal{A}$ is a PPT traceability adversary against our group signature scheme with success probability ε. We construct an algorithm $\mathcal{F}$ that solves the $\mathsf{RSD}(n_0, r_0, w_s)$ problem with success probability polynomially related to ε.

At first, $\mathcal{F}$ receives a challenge from a decisional $2 - \mathsf{IRSD}(n_0, w_s)$ instance, *i.e.,* a random pair $(\overline{\mathbf{H}}, \overline{\mathbf{y}}) \in \mathbb{F}_{q^{m_0}}^{r_0 \times n_0} \times \mathbb{F}_{q^{m_0}}^{r_0}$, where $\overline{\mathbf{H}}$ is an ideal matrix (together with its description $(\mathbf{h}_0, P_0)$). The task of $\mathcal{F}$ is to find a vector $\mathbf{s} \in \mathcal{S}_{w_s}^{n_0, m_0}$ such that $\overline{\mathbf{H}} \cdot \mathbf{s}^T = \overline{\mathbf{y}}^T$. It then proceeds as follows:

1. Pick a guess j^* and set $\mathbf{y}_{j^*} = \overline{\mathbf{y}}$.
2. Set $\mathbf{H} = \overline{\mathbf{H}}$. For each $j \in [N-1] \setminus \{j^*\}$, sample $\mathbf{s}_j \leftarrow \mathcal{S}_{w_s}^{n_0, m_0}$ and set $\mathbf{y}_j = \mathbf{s}_j \cdot \mathbf{H}^T$.
3. Run $\mathsf{RQC.KeyGen}(n, k, w_r)$ to obtain a key pair $(\mathsf{pk}_{\mathsf{RQC}}, \mathsf{sk}_{\mathsf{RQC}})$.
4. Send $\mathsf{gpk} = (\mathbf{H}, \widetilde{\mathbf{H}}, \mathbf{A}, p, P, P_0, F, F_0, \mathbf{M}, \alpha)$ and $\mathsf{gmsk} = \mathsf{sk}_{\mathsf{RQC}}$ to $\mathcal{A}$.

Here, since the decisional $2 - \mathsf{IRSD}(n_0, w_s)$ is hard, so the view of $\mathcal{A}$ on the instance produced by $\mathcal{F}$ is computationally indistinguishable to its view on the instance

from the real protocol. Next, $\mathcal{F}$ responds to the queries from $\mathcal{A}$. It initializes a set $CU = \varnothing$, and proceeds as follows:

1. For queries to the random oracle $\mathcal{H}$, it returns uniformly random values in $\{1,2,3\}^{\kappa}$. Suppose that $\mathcal{A}$ makes $Q_{\mathcal{H}}$ queries to the random oracle, then for each $\eta \leq Q_{\mathcal{H}}$, we let r_{η} be the answer to the η-th query.

2 For query to $\mathcal{O}^{\mathsf{Corrupt}}(j)$, if $j = j^*$, then $\mathcal{F}$ aborts; if $j \neq j^*$, then $\mathcal{F}$ sets $CU := CU \cup \{j\}$ and gives $\mathbf{s}_j$ to $\mathcal{A}$.

3 For query to $\mathcal{O}^{\mathsf{Sign}}(j, M)$, for $j \in [N-1]$ and any message M :

- If $j \neq j^*$, then $\mathcal{F}$ honestly computes a signature by using $\mathbf{s}_j$.
- If $j = j^*$, then $\mathcal{F}$ returns a simulated signature Σ^*.

At some point, $\mathcal{A}$ outputs a forged signature Σ^* on some message M^*, where

$$\Sigma^* = \big(\mathbf{c}^*, \mathsf{CMT}^{(1)}, \ldots, \mathsf{CMT}^{(\kappa)}; \mathsf{Ch}^{(1)}, \ldots, \mathsf{Ch}^{(\kappa)}; \mathsf{RSP}^{(1)}, \ldots, \mathsf{RSP}^{(\kappa)}\big).$$

This signature must satisfy all the requirements of the traceability experiment. Now $\mathcal{F}$ uses $\mathsf{sk}_{\mathsf{RQC}}$ to open Σ^*. It aborts if the opening algorithm does not output j^*. The probability that $\mathcal{F}$ aborts is at most $\frac{N-1}{N} + (\frac{2}{3})^{\kappa}$. Therefore, with probability at least $\frac{1}{N} - (\frac{2}{3})^{\kappa}$, it holds that

$$\begin{cases} \mathsf{Verify}(\mathsf{gpk}, M^*, \Sigma^*) = 1, \\ \mathsf{Open}(\mathsf{sk}_{\mathsf{RQC}}, M^*, \Sigma^*) = j^*. \end{cases}$$

Assume that the above equalities hold, we denote Δ the tuple

$$\big(M^*; \mathsf{CMT}^{(1)}, \ldots, \mathsf{CMT}^{(\kappa)}; \mathbf{H}, \mathbf{H}_1, \ldots, \mathbf{H}_4, \mathbf{y}_1, \ldots, \mathbf{y}_{N-1}, \mathbf{c}^*\big).$$

Observe that with probability at least $p = \varepsilon - 3^{-\kappa}$, there exists a certain $\eta^* \leq Q_{\eta}$ such that Δ was the input of the η^*-th query. $\mathcal{F}$ picks η^* as the target forking point and replays $\mathcal{A}$ many times with the same random tape and input. In each run, for the first $\eta^* - 1$ queries, $\mathcal{A}$ is given the same answers $r_1, \ldots, r_{\eta^*-1}$ as in the initial run; from the η^*-th query onwards, $\mathcal{F}$ answers with fresh random values in $\{1,2,3\}^{\kappa}$. With probability greater than $\frac{1}{2}$ and within $32 \cdot Q_{\mathcal{H}}/p$ executions of $\mathcal{A}$, algorithm $\mathcal{F}$ can obtain a 3-fork, say

$$\begin{aligned} r_{1,\eta^*} &= \big(\mathsf{Ch}_1^{(1)}, \ldots, \mathsf{Ch}_1^{(\kappa)}\big), \\ r_{2,\eta^*} &= \big(\mathsf{Ch}_2^{(1)}, \ldots, \mathsf{Ch}_2^{(\kappa)}\big), \\ r_{3,\eta^*} &= \big(\mathsf{Ch}_3^{(1)}, \ldots, \mathsf{Ch}_3^{(\kappa)}\big). \end{aligned}$$

Note that one has

$$\Pr\left[i \in \{1,\ldots,\kappa\} | \{\mathsf{Ch}_1^{(i)}, \mathsf{Ch}_2^{(i)}, \mathsf{Ch}_3^{(i)}\} = \{1,2,3\}\right] = 1 - \left(\frac{7}{9}\right)^{\kappa}.$$

Conditioned on the existence of such index i, one parses the 3 forgeries corresponding to the fork to obtain $(\mathsf{RSP}_1^{(i)}, \mathsf{RSP}_2^{(i)}, \mathsf{RSP}_3^{(i)})$. They are three valid responses to three different challenges of the same commitment $\mathsf{CMT}^{(i)}$. By using the knowledge extractor as in Lemma 5, one can efficiently find a valid solution to the challenge $\mathsf{RSD}(n_0, r_0, w_s)$ instance $\left(\overline{\mathbf{H}}, \overline{\mathbf{y}}\right)$.

Finally, if $\mathcal{A}$ has success probability ε and running time T in attacking the traceability of our group signature scheme, then $\mathcal{F}$ has success probability at least $\frac{1}{2} \cdot \left(\frac{1}{N} - (\frac{2}{3})^{\kappa}\right) \cdot \left(1 - (\frac{7}{9})^{\kappa}\right)$ and running time $32 \cdot T \cdot Q_{\mathcal{H}}/(\varepsilon - 3^{-\kappa}) + \mathsf{poly}(\lambda, N)$.

4.5 Parameters

In this section, we give a few examples of parameters for our code-based group signature scheme. The parameters are chosen so that the attacks in [AGHT18] and [BBC+20] have complexity at least at level 2^{128} to solve the $\mathsf{RSD}(n_0, r_0, w_s)$ problem or to break the RQC scheme.

- We consider $q = 2$, and thus $\log 2 = 1$. In this case, the map f_1 becomes the representation map with respect to the base 2.
- Parameters for the RQC scheme: $m = 139, n = 101, k = 5$ which are taken from [AAB+17].
- Parameter for the syndrome decoding problem corresponding to the matrix $\mathbf{H}$: $m_0 = 47, r_0 = 43$, $n_0 = 86, w_s = 7$.
- The number of users $N - 1 = q^{\ell} - 1$ for $\ell \in \{4, 8, \ldots, 24\}$.

Recall that the size of public key is

$$T = r_0 mN + (k+2)mn + (\ell + n + r_0 + m_0 + m) + \ell km,$$

and the size of signature is

$$\left(\ell + N - 1 + m_0^2 + m_0 n_0 + n_0^2 + m^2 + 3mn + 3n^2 + km + 5\lambda\right) \cdot \kappa + n.$$

We have the following table ($\kappa = 220$)

ℓ	PK Size	Signatures Size
4	16.71 KB	2.94 MB
8	77.68 KB	2.95 MB
12	1.05 MB	3.06 MB
16	16.57 MB	4.74 MB
20	264.9 MB	31.78 MB
24	4.24 GB	464.3 MB

Table 4.2: Example of parameters.

4.6 Conclusion

In this work, we have constructed a code-based group signature scheme in the rank metric context. In some cases, our parameters are better than those of the previous work as in [ELL+15].

One feature of our scheme may be noteworthy, that is, in the second layer, we made use of RQC, however, one can can use HQC due to the same structure as RQC. The only shortcoming is that the signature size would be larger.

Chapter 5

Blind Signatures from CFS Signatures

This chapter presents a code-based blind signature scheme which is constructed from CFS signatures and Stern's identification protocol. To be more accurate, the scheme here is a correction of the one in [BGSS17].

This is a joint work with Olivier Blazy and Philippe Gaborit and was presented at CBCrypto 2021.

5.1 Introduction

Blind signatures were first introduced by Chaum in 1982 [Cha82]. Unlike usual signatures, the signed message is hidden from the signer (blindness). With this property, blind signatures have found many applications such that electronic voting, electronic cash [Cha82]. There has been many blind signature protocols most of which use the RSA approach [Oka93, PS97] and thus are not considered to be post-quantum secure. The task of constructing post-quantum secure blind signature schemes was first successfully handled by Hauck *et al.* [HKLN20]. Their construction is based on lattice assumptions aided by linear hash functions.

In the code-based field, blind signatures were first considered by Overbeck [Ove09]. Still, the construction has many issues to be reflected on. The second notable attempt was made in 2017, a code-based blind signature scheme was proposed by Blazy *et al.* [BGSS17]. However, there is a flaw in the proof of the unforgeability property due to a lack in the construction. Thus, the proof therein is invalid. The goal of this work is to give a new blind signature scheme based on the one in [BGSS17], which is supported by correct proofs of security.

Organization. The rest of the work is organized as follows. In Section 5.2, we briefly recall some basic notions in code-based cryptography, which are required

for our construction. Section 5.3 describes the security model for our scheme. Section 5.4 recalls the previous scheme and provides the explanation of the flaw in its proof. A new scheme having correct proofs of security and a set of parameters are presented in Section 5.5. Finally, we draw some remarks in Section 5.6.

5.2 Background on Code-Based Cryptography

Let $\mathbb{F}_q$ denote the finite field with q elements, $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ a parity-check matrix of some linear code of length n and dimension k, and $\mathcal{S}_w^n$ the sphere of $\mathbb{F}_q^n$ of radius w. Throughout the chapter, $h(\cdot)$ will stand for a cryptographic hash function.

5.2.1 Syndrome Decoding

Definition of the syndrome decoding problem is already provided in Section 2.4. Beside this problem, another important parameter in code-based cryptography is the Gilbert-Varshamov bound, it is defined as follows.

Definition 5.1. *The volume of a ball of radius w in the* Hamming *space $\mathbb{F}_q^n$ is*

$$V_n(w) = \sum_{i=0}^{w} (q-1)^i \binom{n}{i}.$$

For given n and k, the smallest integer b_{GV} such that $V_n(b_{GV}) \geq q^{n-k}$ is called the Gilbert-Varshamov *(GV) bound.*

Definition 5.2. *We call* w-bounded decoder *associated to $\mathbf{H}$ a procedure $\mathbb{F}_q^{n-k} \to \mathbb{F}_q^n$ which returns for all $\mathbf{u} \in \mathbb{F}_q^{n-k}$ solution of $\mathsf{CSD}(\mathbf{H}, \mathbf{u}, w)$ (or fails if this set is empty).*

For given n and k and for almost all codes, a w-bounded decoder fails for a proportion approximately $\exp(-V_n(w)/q^{n-k})$ of the instances. If we choose an integer $w > b_{GV}$, a w-bounded decoder almost never fails[1]. We will speak of a *complete*[2] decoder.

5.2.2 Trapdoor Digital Signatures

Let w_0 be the smallest integer such that $\mathsf{CSD}(\mathbf{H}, \mathbf{u}, w_0) \neq \varnothing$ with high probability (*i.e.*, from the previous section $w_0 = \lceil b_{gv} \rceil$ or $\lceil b_{gv} + 1 \rceil$). We assume here that

[1] Most of the time $w = b_{GV}$ is enough, exceptionally $w = b_{GV} + 1$.

[2] The word *complete* is used here for convenience, the decoder may fail but for a negligible proportion of the instances.

the linear code defined by the parity check matrix $\mathbf{H}$ has some hidden algebraic structure (for instance a binary Goppa code) which enables a trapdoor complete w_0-bounded decoder $D_{\mathbf{H}}(\cdot)$.

CFS Signatures. Obtaining a practical complete decoder is not an easy task because the desired decoding bound w_0 is above the algebraic error correcting capability. It is possible for binary Goppa codes of high rate (*i.e.,* the ratio k/n between dimension and length is close to 1) [CFS01]: the resulting complete decoder is complex but still has an exponential advantage in complexity compared with the best generic algorithms for solving CSD.

Let $\mathbf{H}$ be the parity check matrix of a CFS code, let $D_{\mathbf{H}}(\cdot)$ be the trapdoor CFS decoding function. The CFS problem is defined as given q accesses to a CFS oracle (given $\mathbf{x}$, it returns $\mathbf{y} = D_{\mathbf{H}}(\mathbf{x})$), and $\mathbf{u}^*$ the adversary has to return $\mathbf{y}^*$ such that $\mathbf{H}\mathbf{y}^{*T} = \mathbf{u}^{*T}$ and $wt(\mathbf{y}^*) = w$ in polynomial time after at most q queries to the oracle, on words different from $\mathbf{u}^*$.

Figure 5.1: The CFS problem.

Security of CFS Signatures. Parity-check matrices of high rate Goppa codes can be distinguished from random matrices [FGO+10]. Still, this distinguishing attack does not lead to an efficient key recovery attack (recovering $D_{\mathbf{H}}$ from $\mathbf{H}$), however it invalidates the security reduction given in [CFS01]. We refer to [LS12] for more details on the security of CFS.

Parallel CFS. It was proposed by Finiasz [Fin11]. It consists in producing λ signatures (3 or 4) of related digests. If done correctly, the cost for an existential forgery attack can be made arbitrarily close to the cost for a universal forgery attack.

5.2.3 Stern's Identication Protocol

This section is dedicated to Stern's idenentication protocol and, specially, to a concatenated Stern authentication protocol. The Stern's identification protocol is already introduced in Section 2.5.2. In the following paragraph, we focus on the latter, which is the randomized version of the one given in [ABCG16a] and will serve as a building-block in our construction.

For ease of notations, from now on, k (and k' in the next sections) will have the equal meaning of co-dimension. Let us consider $\mathbf{Q}$ a $k \times n_1$ binary matrix and $\mathbf{R}$ a $k \times n_2$ binary matrix. Suppose that there exist a vector $(\mathbf{x}, \mathbf{y})$ with $\mathbf{x}, \mathbf{y}$ of respective lengths n_1, n_2 and of weight w_1, w_2, and a syndrome $\mathbf{s}$ such that $[\mathbf{Q}|\mathbf{R}] \cdot (\mathbf{x}, \mathbf{y})^T = \mathbf{s}^T = \mathbf{Q} \cdot \mathbf{x}^T + \mathbf{R} \cdot \mathbf{y}^T$. The (randomized) concatenated Stern

authentication protocol is a zero-knowledge (ZK) protocol which allows the prover $\mathcal{P}$ to prove that he knows a vector $(\mathbf{x}, \mathbf{y})$, for $\mathbf{x}$ and $\mathbf{y}$ of respective weight w_1 and w_2 such that

$$[\mathbf{Q}|\mathbf{R}] \cdot (\mathbf{x}, \mathbf{y})^T = \mathbf{s}^T = \mathbf{Q} \cdot \mathbf{x}^T + \mathbf{R} \cdot \mathbf{y}^T.$$

In the following, S_n denotes the permutation group of length n, and $|$ stands for concatenation. The protocol works as described in Figure 5.2.

Concatenated Stern zero-knowledge authentification protocol
Public data: two matrices $\mathbf{Q}$ and $\mathbf{R}$ of respective size $k \times n_1$ and $k \times n_2$, a syndrome $\mathbf{s}$.
Prover $\mathcal{P}$: a vector $(\mathbf{x}, \mathbf{y})$ for $\mathbf{x}$ and $\mathbf{y}$ of respective weight w_1 and w_2 such that

$$[\mathbf{Q}|\mathbf{R}] \cdot (\mathbf{x}, \mathbf{y})^T = \mathbf{s}^T = \mathbf{Q} \cdot \mathbf{x}^T + \mathbf{R} \cdot \mathbf{y}^T.$$

The prover $\mathcal{P}$ interacts with a verifier $\mathcal{V}$ in 3 steps and a verification:

1. **Commitments.** $\mathcal{P}$ generates $\sigma_1 \leftarrow S_{n_1}$, $\sigma_2 \leftarrow S_{n_2}$, $\mathbf{u}_1 \leftarrow \mathbb{F}_2^{n_1}$,$\mathbf{u}_2 \leftarrow \mathbb{F}_2^{n_2}$ and $r_1, r_2, r_3 \leftarrow 1^\lambda$.

   $\mathcal{P}$ sends three commitments:
   $c_1 = h(\sigma_1(\mathbf{u}_1)|\sigma_2(\mathbf{u}_2)|r_1)$,
   $c_2 = h(\sigma_1|\mathbf{Q}\mathbf{u}_1^T + \mathbf{R}\mathbf{u}_2^T|\sigma_2|r_2)$,
   $c_3 = h(\sigma_1(\mathbf{x} + \mathbf{u}_1)|\sigma_2(\mathbf{y} + \mathbf{u}_2)|r_3)$.

2. **Challenge.** $\mathcal{V}$ responds with $b \in \{0, 1, 2\}$.

3. **Answer.** There are three possibilities:

   - If $b = 0$, P reveals $\sigma_1(\mathbf{u}_1)$, $\sigma_2(\mathbf{u}_2)$, $\sigma_1(\mathbf{x})$, $\sigma_2(\mathbf{y})$, r_1, and r_3.
   - If $b = 1$, he reveals σ_1, σ_2, $\mathbf{x} + \mathbf{u}_1$, $\mathbf{y} + \mathbf{u}_2$, r_2, and r_3.
   - If $b = 2$, he reveals σ_1, $\mathbf{u}_1$, σ_2, $\mathbf{u}_2$, r_1, and r_2.

4. **Verification.**

   - if $b = 0$ checks c_1 and c_3;
   - if $b = 1$ checks c_2 and c_3;
   - if $b = 2$ checks c_1 and c_2.

Figure 5.2: Concatenated Stern zero-knowledge protocol.

Theorem 5.1. *The concatenated* Stern *zero-knowledge protocol is a ZK protocol with cheating probability* $\frac{2}{3}$.

Proof. The protocol we describe is an adaptation of the one described in [ABCG16a] to which we added random values r_1, r_2 and r_3. By doing so, the protocol cannot be testable and leaks no information (see [ABCG16a] for details on testable Stern

protocol). For verification, the only non trivial check is (as for Stern's original protocol) for $b = 1$ and the value c_2, which is checked with the public syndrome $\mathbf{s}$, one recovers $\mathbf{Qu}_1^T + \mathbf{Ru}_2^T$ as $\mathbf{Qu}_1^T + \mathbf{Ru}_2^T = \mathbf{Q}(\mathbf{x} + \mathbf{u}_1)^T + \mathbf{R}(\mathbf{y} + \mathbf{u}_2)^T - \mathbf{s}^T$. The proof is straightforward from [ABCG16a] with the ZK properties obtained from the random values r_1, r_2, and r_3.

□

5.3 Blind Signatures

As formalized by Pointcheval and Stern [PS00], a blind signature scheme involves two parties, a user $\mathcal{U}$ and a signer $\mathcal{S}$. The user submits a masked (or blinded) message that the signer will sign with a digital signature scheme whose public key is known. This part is named $\mathsf{BSProtocol}$. The user unmasks this signature to build a signature of the unmasked message which is valid for the signer's public key. A verification can be made on the final signature with the signer's public key.

More precisely, we can derive the definition of blind signatures from that of digital signatures. Instead of having a signing phase $\mathsf{Sign}(\mathsf{sk}, M; \mu)$, we have an interactive phase $\mathsf{BSProtocol}\langle\mathcal{S}, \mathcal{U}\rangle$ between the user $\mathcal{U}(\mathsf{vk}, M; \rho)$ who will (probably) transmit a masked information on M under some randomness ρ in order to obtain a signature valid under the verification key vk, and the signer $\mathcal{S}(\mathsf{sk}; \mu)$, who will generate something based on this value, and his secret key which should lead the user to a valid signature. Such signatures are correct if when both the user and signer are honest then $\mathsf{BSProtocol}\langle\mathcal{S}, \mathcal{U}\rangle$ does indeed lead to valid signature on M under vk. There are two additional security properties, one protecting the signer, the other the user.

- On one hand, there is an *Unforgeability* property, which states that a malicious user should not be able to compute $n + 1$ valid signatures on different messages after at most n interactions with the signer.

- On the other hand, the *Blindness* property says that a malicious signer who signed two messages M_0 and M_1 should not be able to decide which one was signed first.

These properties are described in Figure 5.3.

$\mathsf{Exp}^{\mathsf{bl}-b}_{\mathcal{BS},\mathcal{S}^*}(\mathfrak{K})$
1. $(\mathsf{param}) \leftarrow \mathsf{BSSetup}(1^{\mathfrak{K}})$
2. $(\mathsf{vk}, M_0, M_1) \leftarrow \mathcal{A}(\texttt{FIND}:\mathsf{param})$
3. $\sigma_b \leftarrow \mathsf{BSProtocol}\langle \mathcal{A}, \mathcal{U}(\mathsf{vk}, M_b)\rangle$
4. $\sigma_{1-b} \leftarrow \mathsf{BSProtocol}\langle \mathcal{A}, \mathcal{U}(\mathsf{vk}, M_{1-b})\rangle$
5. $b^* \leftarrow \mathcal{S}^*(\texttt{GUESS}: M_0, M_1)$;
6. `RETURN` $b^* = b$.

$\mathsf{Exp}^{\mathsf{uf}}_{\mathcal{BS},\mathcal{U}^*}(\mathfrak{K})$
1. $(\mathsf{param}) \leftarrow \mathsf{BSSetup}(1^{\mathfrak{K}})$
2. $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{BSKeyGen}(\mathsf{param})$
3. For $i = 1, \ldots, q_s$, $\mathsf{BSProtocol}\langle \mathcal{S}(\mathsf{sk}), \mathcal{A}(\texttt{INIT}:\mathsf{vk})\rangle$
4. $\big((m_1, \sigma_1), \ldots, (m_{q_s+1}, \sigma_{q_s+1})\big) \leftarrow \mathcal{A}(\texttt{GUESS}:\mathsf{vk})$;
5. `IF` $\exists i \neq j, m_i = m_j$ `OR` $\exists i, \mathsf{Verif}(\mathsf{pk}, m_i, \sigma_i) = 0$
   `RETURN` 0
6. `ELSE RETURN` 1

Figure 5.3: Security games for blind signatures.

In the above games, queries of the adversary are required to be well-formed.

5.4 The Previous Scheme

In this section, we recall the old scheme in [BGSS17] and point out its flaw. The previous scheme is as follows.

$\mathsf{KeyGen}(k, k', n, n')$:
From some integer parameters k, k', n and n', generate:

- $\mathbf{H}$ a trapdoor parity check matrix of size $k \times n$ and its trapdoor $D_{\mathbf{H}}(\cdot)$, only available to the signer $\mathcal{S}$.
- $\mathbf{A}$ a random matrix of size $k \times n'$.
- $\mathbf{B}$ a random matrix of size $k' \times n'$.

Figure 5.4: Key generation.

$\mathsf{BSProtocol}(w_{\mathbf{x}})$:
1) **Blinding step**
The user $\mathcal{U}$

- generates uniformly at random a vector $\mathbf{x}$ in $\mathbb{F}_2^{n'}$ of weight $w_{\mathbf{x}}$.
- sends $\mu = h(M|\mathbf{B}\mathbf{x}^T) + \mathbf{A}\mathbf{x}^T$ to $\mathcal{S}$.

The signer $\mathcal{S}$ returns $\mathbf{y} = D_{\mathbf{H}}(\mu)$ of weight $w_{\mathbf{y}}$ to the user $\mathcal{U}$. If $D_{\mathbf{H}}(\mu)$ returns $\perp$, then $\mathcal{U}$ can send another request.
2) **Blind Signature step**: $\mathcal{U}$ sends the couple $(\mathbf{B}\mathbf{x}^T, \mathsf{PoK})$, where PoK is a transcript of the proof of knowledge that $\mathcal{U}$ knows a pair of vectors $(\mathbf{x}, \mathbf{y})$ of weight $w_{\mathbf{x}}$ and $w_{\mathbf{y}}$ such that

$$\begin{pmatrix} \mathbf{A} & \mathbf{H} \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^T \\ \mathbf{y}^T \end{pmatrix} = \begin{pmatrix} h(M|\mathbf{B}\mathbf{x}^T) \\ \mathbf{B}\mathbf{x}^T \end{pmatrix}.$$

The proof of knowledge is obtained through the concatenated ZK Stern protocol of Section 5.2.3, by taking $\mathbf{Q} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix}$ and $\mathbf{R} = \begin{pmatrix} \mathbf{H} \\ \mathbf{0} \end{pmatrix}$.

Figure 5.5: The blind signature protocol.

Verification: Upon receiving the message M and the signature $\big(\mathbf{B}\mathbf{x}^T, \mathsf{PoK}\big)$, the verifier checks that the proof of knowledge PoK is correct and that the weights $w_{\mathbf{x}}$ and $w_{\mathbf{y}}$ of $\mathbf{x}$ and $\mathbf{y}$ are correct.

Figure 5.6: Verification protocol.

As mentioned above, with this scheme, there is a flaw in the proof of unforgeability property, that is, we can no longer use an adversary, who can break the soundness of the scheme, to solve the CFS problem or break the soundness of the underlying zero-knowledge proof. The reason is that after answering queries for the adversary, the simulator still does not know the values $\mathbf{x}$'s. Thus, two problems follow. First, in the rewinding step, "output another random value" does not guarantee that there are no collisions. Note that the queries to the signing oracle are of the form $h(M|\mathbf{B}\mathbf{x}^T) + \mathbf{A}\mathbf{x}^T$, which also contains the term $\mathbf{A}\mathbf{x}^T$. Second, since the simulator does not know the values $\mathbf{x}$'s, which were used by the adversary, there is no way he can accomplish " setting $h(M_{y_j}|B_j)$ to $\mathbf{u}^* - A_{x_j}$." Thus the adversary could not be used to solve the CFS problem.

5.5 A New Scheme

5.5.1 The Scheme

In this section, we propose a scheme, which corrects the above flaw. The key generation algorithm remains as in the above scheme, the blind and verification protocols are as follows.

$\mathsf{BSProtocol}(w_{\mathbf{x}})$:
1) **Blinding step**
The user $\mathcal{U}$

- generates uniformly at random a vector $\mathbf{x}$ in $\mathbb{F}_2^{n'}$ of weight $w_{\mathbf{x}}$.
- generates $\pi(\mathbf{x})$, a proof of knowledge for $\mathbf{x}$ with respect to $\mathbf{B}\mathbf{x}^T$.
- sends $\mu = h\big(M|\mathbf{B}\mathbf{x}^T|\pi(\mathbf{x})\big) + \mathbf{A}\mathbf{x}^T$ to $\mathcal{S}$.

The signer $\mathcal{S}$ returns $y = D_{\mathbf{H}}(\mu)$ of weight $w_{\mathbf{y}}$ to the user $\mathcal{U}$. If $D_{\mathbf{H}}(\mu)$ returns $\perp$, then $\mathcal{U}$ can send another request.
2) **Blind Signature step**: $\mathcal{U}$ sends the triple $(\mathbf{B}\mathbf{x}^T, \pi(\mathbf{x}), \mathsf{PoK})$, where PoK is a transcript of the proof of knowledge that $\mathcal{U}$ knows a pair of vectors $(\mathbf{x}, \mathbf{y})$ of weight $w_{\mathbf{x}}$ and $w_{\mathbf{y}}$ such that

$$\begin{pmatrix} \mathbf{A} & \mathbf{H} \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}^T \\ \mathbf{y}^T \end{pmatrix} = \begin{pmatrix} h\big(M|\mathbf{B}\mathbf{x}^T|\pi(\mathbf{x})\big) \\ \mathbf{B}\mathbf{x}^T \end{pmatrix}.$$

The proof of knowledge is obtained through the concatenated ZK Stern protocol by taking $\mathbf{Q} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \end{pmatrix}$ and $\mathbf{R} = \begin{pmatrix} \mathbf{H} \\ \mathbf{0} \end{pmatrix}$.

Figure 5.7: The corrected blind signature protocol.

Verification: Upon receiving the message M and the signature $\big(\mathbf{B}\mathbf{x}^T, \pi(\mathbf{x}), \mathsf{PoK}\big)$, the verifier checks that the proofs of knowledge $\pi(\mathbf{x}), \mathsf{PoK}$ are correct and that the weights $w_{\mathbf{x}}$ and $w_{\mathbf{y}}$ of $\mathbf{x}$ and $\mathbf{y}$ are correct.

Figure 5.8: The corrected verification protocol.

In this scheme, we add a proof of knowledge of $\mathbf{x}$ in the hash queries, *i.e.,* a hash query consists of a message M, the value $\mathbf{B}\mathbf{x}^T$, and a Stern-like proof of knowledge $\pi(\mathbf{x})$ of $\mathbf{x}$ with respect to $\mathbf{B}\mathbf{x}^T$. In order to obtain this proof, one has to

query the random oracle three times to get the values of commitments. From these queries, the one in control of the random oracle would certainly know the value $\mathbf{x}$. This argument guarantees that in our proof, the simulator would know the values $\mathbf{x}$'s and with this knowledge, he could efficiently manipulate the random oracle to avoid collisions. Note that $\pi(\mathbf{x})$ only needs to have one round. The verification protocol is considered to be successful if both $\pi(\mathbf{x})$ and PoK are valid.

5.5.2 Unforgeability

Theorem 5.2. *If there exists an adversary against the soundness of the blind signature scheme, then there exists an adversary for either the CFS problem, the syndrome decoding problem, or the soundness of the underlying zero-knowledge proof.*

Proof. If an adversary $\mathcal{A}$ can win the game of unforgeability of the blind signature, then he can produce $N+1$ blind signatures with N requests to the blind oracle. To exploit this adversary, we build a simulator in the following way. We first receive the matrix $\mathbf{H}$ and a hash function h from the challenge oracle for CFS problem and generate normally the other parameter of our blind signature. The hash and signing queries are treated as follows.

- Receiving signing queries, on string $\mathbf{c}_i$, we forward it to the CFS oracle, and receive y_i such that $\mathbf{H}\mathbf{y}_i^T = \mathbf{c}_i^T$.
- Receiving hash queries, the simulator answers with a random value, and stores it to answer in the same way to similar queries.

After at most N signing queries and $\mathfrak{n}$ random oracle queries, the adversary sends us $N+1$ signatures σ_j on messages M_j, by sending us values B_j, π_j, and zero-knowledge proofs PoK_j, that he knows $\mathbf{x}_j, \mathbf{y}_j$ such that $B_j = \mathbf{B}\mathbf{x}_j^T, \pi_j = \pi(\mathbf{x}_j)$, and $\mathbf{H}\mathbf{y}_j^T = h(M_j|B_j|\pi_j) - \mathbf{A}\mathbf{x}_j^T$. As this is a valid forgery against the blind signature scheme, then all the $N+1$ signatures are valid. This means that either the adversary manages to break the soundness of one of the proofs, or by using the random oracle, the simulator manages to extract the values $\mathbf{x}_j, \mathbf{y}_j$.

If two values $\mathbf{y}_{j_1}$ and $\mathbf{y}_{j_2}$ are equal, then the adversary has managed to find a collision on $h(M_{j_b}|B_{j_b}|\pi_{j_b}) - A_{j_b}$, where $A_{j_b} = \mathbf{A}\mathbf{x}_{j_b}^T$. In this case, we simply rewind to the furthest random oracle query on $M_{j_b}|B_{j_b}|\pi_{j_b}$ and output another random value such that there is no longer a collision (neither with the query corresponding to j_b, nor with any queries done before to the random oracle). Note that the queries contain $\pi(\mathbf{x})$, a proof of knowledge of $\mathbf{x}$ with respect to $\mathbf{B}\mathbf{x}^T$. In order to obtain these proofs, the adversary has to query the random oracle on the values of commitments. In this way, the simulator always knows the pairs $(\mathbf{x}, \mathbf{B}\mathbf{x}^T)$, as

long as $\mathcal{A}$ would like to generate $\pi(\mathbf{x})$. Now, the forking lemma ensures us that the adversary's advantages is approximately the same, after k rewinding where k is upper-bounded by $\min(N, \mathfrak{n})$.

After this, we are sure that all the $N+1$ values $\mathbf{y}_j$'s are different, so there exists at least one $\mathbf{y}_j$ that does not come from the challenge oracle. Rewinding one last time, and setting $h(M_{y_j}|B_j|\pi_j)$ to $\mathbf{u}^{*T} - \mathbf{A}\mathbf{x}_j^T$ (this can always be done since the simulator knows the value $\mathbf{x}_j$), and invoking the forking lemmas, allows to recover an $\mathbf{y}_j$ such that $\mathbf{H}\mathbf{y}_j^T = h(\mathbf{u}^*)$ and so it allows to solve the CFS challenge.

□

5.5.3 Blindness

Theorem 5.3. *If there exists an adversary against the blindness of the blind signature, then there exists an adversary under the zero-knowledge property of the* Stern *protocol or the decisional syndrome decoding problem.*

Proof. If an adversary $\mathcal{A}$ can win the game of blindness of the blind signature scheme, then he can break the decisional syndrome decoding problem. To exploit this adversary, we build a simulator in the following way. We first receive a decisional syndrome decoding instance $\mathbf{C}, \mathbf{s}$ and have to guess whether there exists a small $\mathbf{x}$ such that $\mathbf{C} \cdot \mathbf{x}^T = \mathbf{s}^T$. The simulator splits the matrix $\mathbf{C}$ into $\mathbf{A}$ and $\mathbf{B}$ of size $k \times n'$ and $k' \times n'$, respectively (as in the scheme), generates a matrix $\mathbf{H}$ honestly and publishes them as the public keys of the scheme, and gives $\mathbf{H}$'s trapdoor to the adversary. The adversary then sends two messages M_0 and M_1 to the simulator. The simulator picks a random bit $b \leftarrow \{0, 1\}$, and proceeds to send the requests on M_b and M_{1-b}, and then outputs the signature on M_0.

With advantage ϵ, the adversary guesses whether $b = 0$ or not. Next, the simulator proceeds to a sequence of games.

Game G_1. In this game, the simulator proceeds honestly, however, instead of outputting the real $\pi(\mathbf{x}_0)$ and PoK_0, he outputs simulated proofs π_0 and Π_0. At this step, the adversary's view is $\mathbf{B}\mathbf{x}_0^T, \pi_0$, and $h\big(M_b|\mathbf{B}\mathbf{x}_b^T|\pi^*\big) + \mathbf{A}\mathbf{x}_b^T$, where $\pi^* \in \{\pi_0, \pi(\mathbf{x}_b)\}$. We can assume that $\mathbf{A}\mathbf{x}_1^T + h\big(M_1|\mathbf{B}\mathbf{x}_1^T|\pi(\mathbf{x}_1)\big) \neq h\big(M_0|\mathbf{B}\mathbf{x}_0^T|\pi(\mathbf{x}_0)\big) + \mathbf{A}\mathbf{x}_0^T$. (Controlling the random oracle allows to make sure of that, anyway it happens with overwhelming probability.)

Game G_2. In this game, the simulator makes the following change. He splits $\mathbf{s}$ into $\mathbf{s}_1, \mathbf{s}_2$, sets the value of $\mathbf{A}\mathbf{x}_0^T$ to be equal to $\mathbf{s}_1^T$, and the value of $\mathbf{B}\mathbf{x}_0^T$ to be equal to $\mathbf{s}_2^T$.

We analyze the answer of the adversary as follows. If the answer to the challenge was yes, we are still in the previous game G_1. On the contrary, if it was no, it leads us to the last game G_2, where the vector $\mathbf{s}$ does not come from the SD distribution. The last game G_2 yields a completely simulated answer (note

that PoK_0 has already been simulated as Π_0), with random public values, so the adversary has no advantage against the blindness in G_2. The difference between G_2 and G_1 is the decisional syndrome decoding problem, while the zero-knowledge property differentiates G_1 from the real game, *i.e.,* between PoK_0 and Π_0. Hence $\epsilon \leq \mathsf{Adv}_{ZK} + \mathsf{Adv}_{DSD}$. Therefore, there is either an adversary against the DSD problem or the zero-knowledge property of the Stern protocol.

□

5.5.4 Parameters

Overall, the best practical attacks against forgery is the attack against the invertible trapdoor function $D_{\mathbf{H}}(\cdot)$, and the best practical attack for blindness is retrieving a small weight vector $\mathbf{x}$ of weight $w_{\mathbf{x}}$ from the syndrome $\mathbf{B}\mathbf{x}^T$, for a random matrix $\mathbf{B}$. Hence, we choose parameters according to these constraints. The size of the public key is $P = kn + (k + k')n'$. The size of the signature is the total size of $\mathbf{B}\mathbf{x}^T, \pi(\mathbf{x})$, and PoK, which is

$$S = k' + n'(\log n' + 1) + \ell \cdot \big(n(\log n + 1) + n'(\log n' + 1) + 5\lambda\big),$$

where ℓ satisfies $(2/3)^\ell = 2^{-\lambda}$ for λ the security parameter.

We now give example of parameters for our scheme, considering parameters for which a word of weight $w_{\mathbf{x}}$ is unique with very strong probability:

We consider the parallel CFS signature scheme with parameters $n = 2^{18}$, $w_{\mathbf{y}} = 9$ and $k = 162$, $n' = 6000$, $k' = 300$ and $w_{\mathbf{x}} = 30$. For that case, the security of parallel CFS is 2^{82} and 2^{91} for the cost of recovering a unique (with strong probability) $\mathbf{x}$ of weight 30 from its syndromes by matrices $\mathbf{A}$ and $\mathbf{B}$. We choose $\lambda = 80$ and $\ell = 137$ so the size of public key is $P = 5.65$ MB, the size of signature is $S = 86.7$ MB.

5.6 Conclusion

We have proposed a new blind signature scheme to repair the one proposed by Blazy *et al.* [BGSS17]. In general, the size of public key and signature differ only slightly from that of the previous scheme. Only the signature size increases a bit due to the addition of the proof of knowledge of the randomness.

The blinding step of our scheme makes use of the trapdoor function $D_{\mathbf{H}}(\cdot)$ of a CFS signature scheme. It might be tempting to try another primitives such that Durandal [ABG+19] or WAVE [DST19].

Chapter 6

Conclusions and Perspectives

6.1 Conclusions

This thesis presented three contributions to the post-quantum cryptography based on coding theory.

(*i*) The first contribution is a code-based signature scheme in the standard model. We designed a chameleon hash function from classical code-based assumptions. The signature scheme follows the hash-and-sign paradigm in which the constructed function plays the role of hash functions. The security of the scheme is guaranteed by the collision-resistant property of the function and is considered in the standard model.

(*ii*) The second contribution is a group signature scheme in the rank metric context. The scheme consists of three layers: a digital signature scheme in the Stern's frame, the RQC cryptosystem, and a zero-knowledge protocol connecting the first two. Though closely following [ELL+15], the design of permutations of the third layer makes our scheme different from the previous scheme. Moreover, our method can be applied for both rank metric and Hamming metric which seems not to be the case for the method of [ELL+15].

(*iii*) The last contribution is a blind signature scheme which is a corrected version of the one in [BGSS17].

6.2 Perspectives

We conclude this thesis by a sketch for future works.

(*i*) Improving the scheme from the first work, *i.e.,* the chameleon signature scheme. As mentioned in Section 3.6, the chosen parameters for the scheme

were rather raw, and thus, refining these parameters is the task we wish to carry out in the near future.

(*ii*) Constructing chameleon hash function in the rank metric. There are two main challenges (or probably one). The first task is to estimate the probability that a random rank code has the minimum distance at least d; and the second is to estimate the probability that all codewords of a random rank code have weight in a given interval $[t_1, t_2]$. Once these questions are settled, one can derive a KKS-like scheme in the rank metric and hence, a chameleon hash function.

(*iii*) Perfecting the RQC cryptosystem. In all versions of this system up to now [ABD+16, AAB+17, AAB+20], the error vectors share the same support. (In the NIST's 2nd round version, two of the three errors have the same support.) This assumption is also applied for the secret key vectors. The reasons for this condition are due to the formulation of the hardness assumptions and to guarantee the success of decoding, *i.e.,* the word obtained from the knowledge of the secret key and a ciphertext is within the decoding capability of the public code being used. We observe that by using Proposition 2.8, the above condition could be removed. The consequence is that the hardness assumptions should be reformulated.

Bibliography

[AAB+17] Carlos Aguila, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Jean-Cristophe Deneuville, Philippe Gaborit, and Gilles Zémor. Rank quasi cyclic(rqc) first round submission to the nist post-quantum cryptography call, November 2017.

[AAB+20] Carlos Aguila, Nicolas Aragon, Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Maxime Bros, Alain Couvreur, Jean-Cristophe Deneuville, Philippe Gaborit, Adrien Hautville, and Gilles Zémor. Rank quasi cyclic(rqc) second round submission to the nist post-quantum cryptography call, 2020.

[ABCG16a] Quentin Alamélou, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. A code-based group signature scheme. Cryptology ePrint Archive, Report 2016/1119, 2016. https://eprint.iacr.org/2016/1119.

[ABCG16b] Quentin Alamélou, Olivier Blazy, Stéphane Cauchie, and Philippe Gaborit. A practical group signature scheme based on rank metric. In *Arithmetic of Finite Fields*, pages 258–275. Springer International Publishing, 2016.

[ABD+16] Carlos Aguilar, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. Efficient encryption from random quasi-cyclic codes. Cryptology ePrint Archive, Report 2016/1194, 2016. https://eprint.iacr.org/2016/1194.

[ABG+19] Nicolas Aragon, Olivier Blazy, Philippe Gaborit, Adrien Hauteville, and Gilles Zémor. Durandal: A rank metric based signature scheme. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 728–758, Darmstadt, Germany, May 19–23, 2019. Springer, Heidelberg, Germany.

[AGHT18] Nicolas Aragon, Philippe Gaborit, Adrien Hautville, and Jean-Pierre Tillich. A new algorithm for solving the rank syndrome decoding

problem. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 2421–2425, 2018.

[Ajt96] Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108, Philadephia, PA, USA, May 22–24, 1996. ACM Press.

[Ale03] Michael Alekhnovich. More on average case vs approximation complexity. In *44th FOCS*, pages 298–307, Cambridge, MA, USA, October 11–14, 2003. IEEE Computer Society Press.

[BBB+21] Slim Bettaieb, Loïc Bidoux, Olivier Blazy, Yann Connan, and Philippe Gaborit. A gapless code-based hash proof system based on rqc and its applications. Cryptology ePrint Archive, Report 2021/026, 2021. https://eprint.iacr.org/2021/026.

[BBC+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In *Advances in Cryptology – ASIACRYPT 2020*, pages 507–536. Springer International Publishing, 2020.

[BCMN11] Paulo S. L. M. Barreto, Pierre-Louis Cayrel, Rafael Misoczki, and Robert Niebuhr. Quasi-dyadic CFS signatures. In *Information Security and Cryptology*, pages 336–349. Springer Berlin Heidelberg, 2011.

[BGSS17] Olivier Blazy, Philippe Gaborit, Julien Schrek, and Nicolas Sendrier. A code-based blind signature. *IEEE International Symposium on Information Theory*, 2017.

[BJMM12] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 520–536, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[BKKP15] Olivier Blazy, Saqib A. Kakvi, Eike Kiltz, and Jiaxin Pan. Tightly-secure signatures from chameleon hash functions. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 256–279, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Heidelberg, Germany.

[BLP11] Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In Phillip Rogaway, editor, *CRYPTO 2011*, volume 6841 of *LNCS*, pages 743–760, Santa Barbara, CA, USA, August 14–18, 2011. Springer, Heidelberg, Germany.

[BMS11] Paulo S.L.M. Barreto, Rafael Misoczki, and Marcos A. Simplicio Jr. One-time signature scheme from syndrome decoding over generic error-correcting codes. *Journal of Systems and Software*, 84(2):198–204, 2011.

[BMvT78] Elwyn Berlekamp, Robert McEliece, and Henk van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 614–629, Warsaw, Poland, May 4–8, 2003. Springer, Heidelberg, Germany.

[BS07] Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 201–216, Beijing, China, April 16–20, 2007. Springer, Heidelberg, Germany.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 157–174, Gold Coast, Australia, December 9–13, 2001. Springer, Heidelberg, Germany.

[Cha82] David Chaum. Blind signatures for untraceable payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *CRYPTO'82*, pages 199–203, Santa Barbara, CA, USA, 1982. Plenum Press, New York, USA.

[Che96] Kefei Chen. A new identification algorithm. In *Cryptography: Policy and Algorithms*, pages 244–249. Springer Berlin Heidelberg, 1996.

[DST19] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new family of trapdoor one-way preimage sampleable func-

tions based on codes. In Steven D. Galbraith and Shiho Moriai, editors, *ASIACRYPT 2019, Part I*, volume 11921 of *LNCS*, pages 21–51, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany.

[DT18] Thomas Debris-Alazard and Jean-Pierre Tillich. Two attacks on rank metric code-based schemes: RankSign and an IBE scheme. In Thomas Peyrin and Steven Galbraith, editors, *ASIACRYPT 2018, Part I*, volume 11272 of *LNCS*, pages 62–92, Brisbane, Queensland, Australia, December 2–6, 2018. Springer, Heidelberg, Germany.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18, Santa Barbara, CA, USA, August 19–23, 1984. Springer, Heidelberg, Germany.

[ELL+15] Martianus Frederic Ezerman, Hyung Tae Lee, San Ling, Khoa Nguyen, and Huaxiong Wang. A provably secure group signature scheme from code-based assumptions. In Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 260–285, Auckland, New Zealand, November 30 – December 3, 2015. Springer, Heidelberg, Germany.

[FGO+10] Jean-Charles Faugère, Valérie Gauthier, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate McEliece cryptosystems. Cryptology ePrint Archive, Report 2010/331, 2010. https://eprint.iacr.org/2010/331.

[Fin11] Matthieu Finiasz. Parallel-CFS - strengthening the CFS McEliece-based signature scheme. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *SAC 2010*, volume 6544 of *LNCS*, pages 159–170, Waterloo, Ontario, Canada, August 12–13, 2011. Springer, Heidelberg, Germany.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.

[FS09] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In Mitsuru Matsui, editor, *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105, Tokyo, Japan, December 6–10, 2009. Springer, Heidelberg, Germany.

[Gab05] Philippe Gaborit. Shorter keys for code-based cryptography. In *Proceedings of International Workshop on Coding and Cryptography WCC'05*, pages 81–91, 2005.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing - STOC '85*. ACM Press, 1985.

[GMRZ13] Philippe Gaborit, Gaétan Mutat, Olivier Ruratta, and Gilles Zémor. Low rank parity check codes and their application to cryptography. In *Proceedings of the Workshop on Coding and Cryptography WCC'13*, 2013.

[Gol87] Oded Goldreich. Two remarks concerning the Goldwasser-Micali-Rivest signature scheme. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 104–110, Santa Barbara, CA, USA, August 1987. Springer, Heidelberg, Germany.

[Gol01] Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Richard E. Ladner and Cynthia Dwork, editors, *40th ACM STOC*, pages 197–206, Victoria, BC, Canada, May 17–20, 2008. ACM Press.

[GRS16] Philippe Gaborit, Olivier Ruatta, and Julien Schrek. On the complexity of the rank syndrome decoding problem. *IEEE Transactions on Information Theory*, 62(2):1006–1019, 2016.

[GSZ11] Philippe Gaborit, Julien Schrek, and Gilles Zémor. Full cryptanalysis of the chen identification protocol. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 35–50, Tapei, Taiwan, November 29 – December 2 2011. Springer, Heidelberg, Germany.

[GZ16] Philippe Gaborit and Gilles Zémor. On the hardness of the decoding and the minimum distance problems for rank codes. *IEEE Transactions on Information Theory*, 62(12):7245–7252, 2016.

[HKLN20] Eduard Hauck, Eike Kiltz, Julian Loss, and Ngoc Khanh Nguyen. Lattice-based blind signatures, revisited. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171

of *LNCS*, pages 500–529, Santa Barbara, CA, USA, August 17–21, 2020. Springer, Heidelberg, Germany.

[KKS97] Gregory Kabatianskii, E. Krouk, and Ben J. M. Smeets. A digital signature scheme based on random error-correcting codes. In Michael Darnell, editor, *6th IMA International Conference on Cryptography and Coding*, volume 1355 of *LNCS*, pages 161–167, Cirencester, UK, December 17–19, 1997. Springer, Heidelberg, Germany.

[KKS05] Grigorii Kabatiansky, Evgenii Krouk, and Sergei Semenov. *Error Correcting Coding and Security for Data Networks: Analysis of the Superchannel Concept*. John Wiley & Sons, 2005.

[KR00] Hugo Krawczyk and Tal Rabin. Chameleon signatures. In *NDSS 2000*, San Diego, CA, USA, February 2–4, 2000. The Internet Society.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 372–389, Melbourne, Australia, December 7–11, 2008. Springer, Heidelberg, Germany.

[LLNW14] Adeline Langlois, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-based group signature scheme with verifier-local revocation. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 345–361, Buenos Aires, Argentina, March 26–28, 2014. Springer, Heidelberg, Germany.

[LN94] Rudolf Lidl and Harald Niederreiter. *Introduction to Finite Fields and their Applications*. Cambridge University Press, 1994.

[LNSW13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In Kaoru Kurosawa and Goichiro Hanaoka, editors, *PKC 2013*, volume 7778 of *LNCS*, pages 107–124, Nara, Japan, February 26 – March 1, 2013. Springer, Heidelberg, Germany.

[Loi06] Pierre Loidreau. Properties of codes in rank metric. *CoRR*, abs/cs/0610057, 10 2006.

[LS12] Gregory Landais and Nicolas Sendrier. Implementing CFS. In Steven D. Galbraith and Mridul Nandi, editors, *INDOCRYPT 2012*,

volume 7668 of *LNCS*, pages 474–488, Kolkata, India, December 9–12, 2012. Springer, Heidelberg, Germany.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 738–755, Cambridge, UK, April 15–19, 2012. Springer, Heidelberg, Germany.

[McE78] Robert J. McEliece. A public-key cryptosystem based on algebraic coding theory. The deep space network progress report 42-44, Jet Propulsion Laboratory, California Institute of Technology, January/February 1978. https://ipnpr.jpl.nasa.gov/progress_report2/42-44/44N.PDF.

[MR04] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on Gaussian measures. In *45th FOCS*, pages 372–381, Rome, Italy, October 17–19, 2004. IEEE Computer Society Press.

[MTSB12] Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. Cryptology ePrint Archive, Report 2012/409, 2012. https://eprint.iacr.org/2012/409.

[NZZ15] Phong Q. Nguyen, Jiang Zhang, and Zhenfeng Zhang. Simpler efficient group signatures from lattices. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 401–426, Gaithersburg, MD, USA, March 30 – April 1, 2015. Springer, Heidelberg, Germany.

[Oka93] Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 31–53, Santa Barbara, CA, USA, August 16–20, 1993. Springer, Heidelberg, Germany.

[OT11] Ayoub Otmani and Jean-Pierre Tillich. An efficient attack on all concrete KKS proposals. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011*, pages 98–116, Tapei, Taiwan, November 29 – December 2 2011. Springer, Heidelberg, Germany.

[Ove09] Raphael Overbeck. A step towards QC blind signatures. Cryptology ePrint Archive, Report 2009/102, 2009. https://eprint.iacr.org/2009/102.

[Pra62] Eugene Prange. The use of information sets in decoding cyclic codes. *IEEE Transactions on Information Theory*, 8:5–9, 1962.

[PS97] David Pointcheval and Jacques Stern. New blind signatures equivalent to factorization (extended abstract). In Richard Graveman, Philippe A. Janson, Clifford Neuman, and Li Gong, editors, *ACM CCS 97*, pages 92–99, Zurich, Switzerland, April 1–4, 1997. ACM Press.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, June 2000.

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93, Baltimore, MA, USA, May 22–24, 2005. ACM Press.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, feb 1978.

[Sho94] Peter W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th FOCS*, pages 124–134, Santa Fe, NM, USA, November 20–22, 1994. IEEE Computer Society Press.

[Sin64] Richard Singleton. Maximum distance q-nary codes. *IEEE International Symposium on Information Theory*, 10(2):116–118, 1964.

[Ste89] Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications*, pages 106–113. Springer-Verlag, 1989.

[Ste94] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 13–21, Santa Barbara, CA, USA, August 22–26, 1994. Springer, Heidelberg, Germany.