A Multi-Signature Scheme based on Coding Theory

Mohammed Meziani and Pierre-Louis Cayrel
CASED–Center for Advanced Security Research Darmstadt
Mornewegstrasse 32, 64293 Darmstadt, Germany
Email: {mohammed.meziani, pierre-louis.cayrel@cased.de}

Abstract—In this paper we propose two first non-generic constructions of multisignature scheme based on coding theory. The first system make use of the CFS signature scheme and is secure in random oracle while the second scheme is based on the KKS construction and is a few times. The security of our construction relies on a difficult problems in coding theory: The Syndrome Decoding problem which has been proved NP-complete [4].

Keywords—Post-quantum cryptography, Coding-based cryptography, Digital signature, Multisignature scheme.

I. Introduction

Digital signature schemes, similar to handwritten signatures, are a fundamental cryptographic primitive used in practice for authenticity and non-repudiation of messages. Several signature schemes exist, but most of them are based on the computational difficulty of solving number theoretic problems such factoring problem, discrete logarithm problem in the multiplicative group of a prime field or in the group of points of an elliptic curve over a finite field. But, in the event of quantum computers all these schemes could be broken due to Shor's algorithm [29] proposed in 1997. Indeed, the Shor's algorithm can solve both the factoring problem and the discrete log problem in finite filelds and on elliptic curves in polynomial time. Therefore, the cryptographic community has to investigate other mathematical problems that are believed to be hard to solve by quantum algorithms. Among these there are problems in coding theory using error correcting codes. The problem of decoding general codes is such a problem, which has been proven to be NP-compelete by Berlekamp, McEliece and Van Tilborg [4].

In 1978, McEliece [22] first proposed an asymmetric cryptosystem which is based on the coding theory and derives its security from the general decoding problem. No efficient attack on this schemes has been found up to date, though numerous computationally intensive attacks have been published in the literature [5], [12]. The idea behind this scheme is to first select a particular (linear) code for which an efficient decoding algorithm is known, and then to use a trapdoor function to disguise the code as a general linear code.

The encryption in the McEliece cryptosytem is not invertible, and therefore it cannot be used for authentication or signature schemes, this is indeed why very few signature schemes based on coding theory have been proposed. This problem was open until 2001 in when Courtois et.al [9] showed how to achieve a code-based signature scheme whose security is based on the syndrome decoding problem. While this problem is NP-complete, their construction is still inefficient for large numbers of errors. Recently, a few code-based signature schemes with additional properties have been published and most of them make use the construction proposed in [9].

ID-based cryptography. The motivation behind the identity based cryptography, proposed by Shamir in 1984 [28], was to simplify the PKI requirements. Instead of using the public key, a user can use his identity (e.g. e-mail address or IP-address) while the associated secret key can be issued by a trusted key generation center (KGC) thanks to a master secret key that only the KGC knows. And thereby some of the costs associated to PKI and certificates can be avoided. Despite this, the identity-based cryptography suffers from a major drawback since a complete trust must be placed on the KGC. This problem is known as the key escrow problem. To overcome this problem, a solution has been proposed in [6] which consists in employing multiple KGCs to jointly produce the master secret key.

Multisignature schemes. A multisignature scheme (MSS) is a normal signature scheme that enables a group of users to cooperatively sign the same document and can be verified by any user. Multisignature schemes have many practical uses such as signing legal electronic documents (e.g. contracts, cheque, etc) by multiple managers in a company. Based on the nature of the application scenarios, the multisignature schemes are divided into categories depending on the signing manner: serial and parallel signing. In the first case, the resulting multisignature is equal to the signature generated by the last signer. More precisely, a signer produces his own signature on a document then broadcasts it to the next signer which after verifying it signs the received components and so on. Here the signing order property should be taken into account. That is, the resulting multisignature depends on the signing order. In the second case the multisignature is produced by a designated signer, called a clerk, which has to collect individual signatures generated by each signer and then combine them into a single signature.

Multisignature schemes have been first introduced in [16]. However, these schemes have an efficiency issue because the generation and the verification cost of the multisignature increases linearly with the number of signers. Since then, various multisignature schemes have been realized. For example, multisignature schemes that are based on RSA assumption [15], [14], [26], constructed form bilinear maps

[7], [30], based on DL assumption [13], [2] and derived from identification schemes like the Fiat-Shamir [27].
With regard to security, the author of [23] provided the first formal security model of multisignature schemes called *Accountable-subgroup multisignatures*. In this model, a provably secure multisignature scheme has to satisfy two important properties: flexibility and accountability. The first property guarantees that any subset of signers can jointly produce a signature on a document and any verifier can decide whether this subset was sufficient to accept the signature while the second property ensures that the identity of any signer can be revealed from the signed document without a trusted third party. This property is very interesting in the sense that if an incorrectly issued multisignature is detected, then it is necessary to identify the corrupted signer. Moreover, this model assumes that the set of signers is known a priori and a signer is not allowed to generate own partial signature before the previous one has been completed. Following this model, [18] proposed multisignature schemes based on the probabilistic signature scheme while [25] designed multisignature schemes using the full domain hash. In these constructions, the signing order is performed in a serial manner and the length of signature as well as the signing cost grows with the number of signers.

Recently, provably secure multisignature schemes using trapdoor one-way permutations [20], [19] have been proposed. These schemes make use of the probabilistic full domain hash and the probabilistic signature scheme, respectively and they are both tightly secure in the random oracle model. Furthermore, the key length in these schemes is independent from the signing order and the length of the signature increases by 30 bits per a signer.

Our contribution:
In this paper, we propose two serial multisignature schemes using error correcting codes. To the best of our knowledge there is no existing multisignature schemes based on coding theory. We use the modified version of CFS signature scheme [9] and the KKS signature scheme [17] as the base of our multisignature schemes. These schemes are secure against existential forgery under adaptive chosen message attack in the random oracle model assuming computational syndrome decoding problem is hard. The first scheme achieves a signature size of $377 + 18.47N$ bits for a security level of $2^{81.5}$, where N is the number of signers. The second scheme produces signatures whose length is independent of N. For instance, 1873.8 bits for a security level of $2^{80.22}$. However, both systems require large public keys of size 0.7 MB and 0.13 MB, respectively.

Organisation:
After recalling some basic definitions and hard problems in coding theory in Section II, we list two code-based signature schemes that we need in our constructions in Section III. In Section IV, we present our code-based multisignature schemes, and we conclude in Section V.

II. Coding Theory Background

This section will first provide a brief introduction to coding theory, then give the basic definitions and list some hard problems we use throughout this paper.

A. Coding theory

The term coding theory refers to a broad branch of mathematics concerned with transmitting data across noisy channels and recovering the message. It provides secure transmission of messages, in the sense that any errors which are introduced during the transmission can be corrected.

B. Notations and Definitions

Let $\mathbb{F}_q$ to denote the finite field with q elements.

a) Codes: An (n,k)-code over $\mathbb{F}_q$ is a linear subspace $\mathcal{C}$ of the linear space $\mathbb{F}_q^n$. Elements of $\mathbb{F}_q^n$ are called *words* and elements of $\mathcal{C}$ are *codewords*. We call n the *length*, and k the *dimension* of the code. If $q = 2$, the code is called *binary*, and is denoted by $[n,k]$.

b) Hamming distance, Hamming weight: The *Hamming distance* $d(x,y)$ between two words $x, y \in \mathbb{F}_q^n$ counts the number of positions in which x and y differ. More formally, denote $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$. Then $d(x,y) = |\{i \ : \ x_i \neq y_i\}|$. Here, we use $|S|$ to denote the number of elements, or cardinality, of a set S. The *Hamming weight* (or just *weight*) of a word $x \in \mathbb{F}_q^n$ is denoted by $wt(x)$ and represents the number of non-zero entries of this word, i.e., $wt(x) = d(\mathbf{0}, x)$, where $\mathbf{0}$ is the vector containing n 0's.

c) Minimum distance: The *minimum distance* d of an (n,k)-code $\mathcal{C}$ is the minimum Hamming distance between two codewords, i.e., $d = \min_{x,y \in C,\ x \neq y} d(x,y)$.

d) Generator matrix, systematic codes: A *generator matrix* of an (n,k)-linear code $\mathcal{C}$ is a $k \times n$ matrix G whose rows form a basis for the vector subspace $\mathcal{C}$, i.e., $\mathcal{C} = \{xG \ : \ x \in \mathbb{F}_q^k\}$. Notice that $\mathcal{C}$ is not unique for a code $\mathcal{C}$. We call a code *systematic* if it can be characterized by a generator matrix $\mathcal{C}$ of the form $G = (I_{k\times k}|A_{k\times(n-k)})$, where $I_{k\times k}$ is the $k \times k$ identity matrix and A an $k \times (n-k)$ matrix.

e) Parity-check matrix, dual code: A *parity-check* matrix of an (n,k)-linear code $\mathcal{C}$ is an $(n-k) \times n$ matrix H whose rows form a basis of the orthogonal complement of the vector subspace $\mathcal{C}$, i.e. it holds that, $C = \{c \in \mathbb{F}_q^n \ : \ Hc^T = \mathbf{0}\}$. Note that H can be viewed as the generator matrix of an $(n, n-k)$ linear code $\mathcal{C}^\perp$ containing codewords $\tilde{c}$ such that for all codewords $c \in \mathcal{C}$, it holds that $\tilde{c}^T.c = 0$. The $\mathcal{C}^\perp$ is generally referred to as the *dual* code of $\mathcal{C}$.

f) Syndrome: Let H be a parity check matrix of the code $\mathcal{C}$. The syndrome of a word $x \in \mathbb{F}_q^n$ is a vector $s \in \mathbb{F}_q^{n-k}$ defined by $s = Hx^T$.

C. Hard problems

In what follows, we recall some hard problems. The security of most code-based cryptosystems is related to hardness of solving these problems.

1) Syndrome Decoding problem (SD):

- Input: An $r \times n$ matrix H over $\mathbb{F}_q$, a target vector $s \in \mathbb{F}_q^r$ and an integer $t > 0$.
- Question: Is there a vector $x \in \mathbb{F}_q^n$ of weight $\leq t$, such that $s = Hx^T$?.

This problem has been proved to be NP-complete by Berlekamp, McEliece, and van Tilborg [4] in 1978 for the general class of binary linear codes. In 1994, Barg [1] extended this result over linear codes defined over $\mathbb{F}_q$. NP-completeness ensures that this problem can not be solved in polynomial time in the worse case, meaning that there are some hard instances, not that every instance is hard.

To end this section, we state another hard problem, Goppa Parametrized Bounded Decoding problem (GPBD), which is a variation of SD problem and have been proved to be NP-complete by Finiasz [11] in 2004.

2) Goppa Parametrized Bounded Decoding (GPBD):

- Input: An $(n-k) \times n$ matrix H over $\mathbb{F}_2$ and a syndrome $s \in \mathbb{F}_2^{n-k}$
- Question: A word $x \in \mathbb{F}_2^n$ of weight $\leq \frac{n-k}{\log_2(n)}$, such that $Hx^T = s$?.

III. The underlying code-based signature schemes

Our constructions are based on two code-based signature schemes that are the Courtois *et al.'s* signature (CFS) [9] and the Kabatianskii *et al.'s* signature scheme (KKS) [17]. Here is the description of two schemes.

A. CFS Signature Scheme

1) Description: For a long time no code-based signature scheme was known, until the first (unbroken) was proposed by Courtois, Finiasz and Sendrier [9] (CFS) in 2001. The basic idea of the CFS signature scheme is to choose parameters such that an inversion for the Niederreiter scheme is practically possible. This is done at the cost of rather large parameters (except for the length of the signature) when comparing to other signature schemes, but at least it does exist !. Before describing the CFS scheme we first recall the Niederreiter public key cryptosystem in Algorithm 1.

Algorithm 1 The Niederreiter PKC

Key Generation:
- Consider an (n,k)-code $\mathcal{C}$over $\mathbb{F}_q$ having a decoding algorithm γ.
- Construct an $(n-k) \times n$ parity check matrix H of $\mathcal{C}$.
- Choose randomly an $(n-k) \times (n-k)$ invertible matrix Q over $\mathbb{F}_q$.
- Choose randomly an $n \times n$ permutation matrix P over $\mathbb{F}_q$.
- Set $\widetilde{H} = PHQ$ as public, and (P, H, Q, γ) as secret.

Encryption: To encrypt a message $x \in \mathbb{F}_q^n$ of weight t
- Compute $y = \widetilde{H}x^T$.

Decryption: To decrypt a cipher $y \in \mathbb{F}_q^{n-k}$ s.t. $y = \widetilde{H}x^T$
- Compute $Q^{-1}y$ $(= HPx^T)$
- Find Px^T from $Q^{-1}y$ by applying γ
- Find x by applying P^{-1} to Px^T.

The McEliece or the Niederreiter schemes are not naturally invertible, i.e. if one starts from a random element y of $\mathbb{F}_2^n$ and a code $\mathcal{C}[n,k,d]$ capable of correcting $\frac{d-1}{2}$ errors, it is almost sure that we won't be able to decode y into a codeword of $\mathcal{C}$. This comes from the fact that the density of decodable words is very small.

Courtois, Finiasz and Sendrier proposed in [9] the first practical signature scheme based on coding theory. The Full Domain Hash (FDH) approach assumes that all the hash values can be inverted by decryption.

The CFS signature scheme is based on the Niederreiter cryptosystem: signing a document requires to hash it into a syndrome and then to try to decode this syndrome. However, for a t-error correcting Goppa code of length $n = 2^m$, only a fraction of approximately $1/t!$ of the syndromes are decodable. Thus, a counter is appended to the message and the signer tries successive counter values until the hash value is decodable. The signature consists of both the error pattern of weight t corresponding to the syndrome, and the value of the counter giving this syndrome.

Algorithm 2 The CFS signature

Key Generation:
- Pick a random parity check matrix H of a (n,k)-binary Goppe code correcting up to t errors and having a decoding algorithm γ.
- Construct the matrices Q, $\widetilde{H}$ and P as in Algorithm 1.

Signature: To sign a message m
(1) $i \leftarrow i + 1$
(2) $x' = \gamma\left(Q^{-1}h(m\|i)\right)$
(3) if no x' was found go to 1
- Output $(i, x'P)$

Verification:
- Compute $s' = Hx'^T$ and $s = h(m\|i)$.
- The signature is valid if s and s' are equals.

2) Security: In [12], the authors present an attack against the CFS scheme due to Daniel Bleichenbacher. Due to this attack, the values of m and t used in the CFS scheme have to change. The authors of [12] propose $m = 21$ and $t = 10$, or $m = 19$ and $t = 11$, or $m = 15$ and $t = 12$, as new parameters for a security of more than 2^{80} binary operations. Due to this attack, the values of m and t used in the CFS scheme have to change. The authors of [12] propose $m = 21$ and $t = 10$, or $m = 19$ and $t = 11$, or $m = 15$ and $t = 12$, as new parameters for a security of more than 2^{80} binary operations.

3) Security proof in the random oracle model: In [10], the author proposes to choose this counter randomly in $\{1, \ldots, 2^{n-k}\}$, and then obtain a proof of security in the random oracle model.

B. KKS signature scheme

Kabatianskii *et al.* [17] proposed a signature scheme based on arbitrary linear error-correcting codes. Actually, they proposed to use a *linear* application f. Three versions are given which are presented in the sequel but all have one point in common: for any $m \in \mathbb{F}_q^k$, the signature $f(m)$ is a codeword

of a linear code $\mathcal{U}$. Each version of KKS proposes different linear codes in order to improve the scheme. We now give a full description of their scheme.

1) Description: Firstly, we suppose that $\mathcal{C}$ is defined by a random parity check matrix H. We also assume that we have a very good estimate d of its minimum distance.Next, we consider a linear code $\mathcal{U}$ of length $n' \leq n$ and dimension k defined by a generator matrix $G = [g_{i,j}]$. We suppose that there exist two integers t_1 and t_2 such that $t_1 \leq \mathsf{w}(u) \leq t_2$ for any non-zero codeword $u \in \mathcal{U}$.

Let J be a subset of $\{1, \ldots, n\}$ of cardinality n', $H(J)$ be the sub matrix of H consisting of the columns h_i where $i \in J$ and define an $r \times n'$ matrix $F \stackrel{\text{def}}{=} H(J)G^T$. The application $f : \mathbb{F}_q^k \rightarrow \mathscr{M}_{n,t}$ is then defined by $f(m) = mG^*$ for any $m \in \mathbb{F}_q^k$ where $G^* = [g^*_{i,j}]$ is the $k \times n$ matrix with $g^*_{i,j} = g_{i,j}$ if $j \in J$ and $g^*_{i,j} = 0$ otherwise. The public application χ is then $\chi(m) = Fm^T$ because $HG^{*T} = H(J)G^T$. The main difference with Niederreiter signatures resides in the verification step where the receiver checks that:

$$t_1 \leq \mathsf{w}(z) \leq t_2 \quad \text{and} \quad F \cdot m^T = H \cdot z^T.$$

Algorithm 3 The KKS signature

Key Generation:
- Select two positive integers t_1 and t_2 s.t. $t_1 \leq t_2$.
- Pick a random parity check matrix $H = [I_r|D]$ of an $(n, n-r)$-code.
- Construct the matrices Q, $\widetilde{H}$ and P as in Algorithm 1.

Signature: To sign a message m
(1) $i \leftarrow i + 1$
(2) $x' = \gamma\left(Q^{-1}h(m\|i)\right)$
(3) if no x' was found go to 1
- Output $(i, x'P)$

Verification:
- Compute $s' = Hx'^T$ and $s = h(m\|i)$.
- The signature is valid if s and s' are equals.

It has been proved in [8], that this scheme is few times.

IV. Our proposed serial multisignature schemes

Before presenting our constructions, we give first the formal definition of a multisignature scheme. We denote by $\mathcal{S} = \{S_1, \ldots, S_N\}$ the set of N signers intended to sign the message M.

A. Definition

A multisignature scheme $\mathcal{MS}$ consists of three algorithms: the key generation $\mathcal{MK}$, the mutisignature generation $\mathcal{MS}$ and the multisignature verification $\mathcal{MV}$) that are defined as follows:

- $\mathcal{MK}$ takes a security parameter and returns a public/secret key pair (pk_i, sk_i) for a signer S_i.
- $\mathcal{MS}$ takes the set of secert keys (sk_i) and a message M and outputs a common a multisignature σ.
- $\mathcal{MV}$ takes the set of public keys (pk_i) (or only one public key), a multisignature σ and the message M and outputs 1 (acceptes) or 0 (rejectes).

The proposed serial multisignature schemes here follow the model of [23] which requires a priori knowledge of an ordered signers set $\{S_1, \ldots S_N\}$. The basic idea of our multisignature schemes is that a signer S_i first generates a signature σ_i on a message M and broadcasts it to the next signer S_{i+1} for further processing. After verifying σ_i, S_{i+1} produces a valid signature on the received components. The generation of the multisignature will be complete when the last signer S_N signs the message.

B. CFS-based serial multisignature

1) Description: Our scheme can be regarded as the extended version of the modified CFS algorithm [10]. In this scheme a signer S_i makes use of the CFS signature decoding algorithm to generate its signature based on the previous signature produced by the signer S_{i-1}. Before the signing step, all signers first collaborate to produce a public random vector r in a serial manner which will be signed together with the message M. In order to check the validity of the resulting multisignature, only the public key of the last signer in the queue will be needed. The CFS multisignature scheme is illustrated in Algorithm 4.

2) Performance Analysis: Using an $(2^m, 2^m - mt)$ Goppa code, each public key H_i is a binary matrix of size $mt \times 2^m$ bits which takes about 99 Mbytes for $t = 9$ and $m = 22$, the multisignature generation consists in producing of N successive CFS signatures of each signer, each of them requires $t^2m^3t!$ binary operations, where N is the number of signers. Verification requires one matrix-vector multiplication and N hash computing. A matrix-vector multiplication can be performed in approximately $t2^m$ binary operations using the mailman algorithm [21]. The CFS-multisignature is composed of a vector of $\mathbb{F}_2^{2^m}$ of weight less than t, N indexes from $\{1, \ldots, 2^{tm}\}$ and a vector of $\mathbb{F}_2^{tm}$. Thus the size of CFS-multisignature is bounded by $\lfloor \log_2 \binom{2^m}{t} \rfloor + N \log_2(t!) + tm$.

We can easily see that the performance evaluation of the proposed multisignature scheme depends mainly on the choice of parameters m and t. If we want to get a reasonable signature cost, we will need a t not greater than 10, for example $(m, t) = (22, 9)$ that give a security level of $2^{81.7}$ according [12]. But if we want to minimize the public key size as well as the signature length, we take $(m, t) = (15, 12)$ for a security level of $2^{81.5}$ [12]. In this case, the signature length amounts to $377 + 18.47N$ bits.

3) Security Analysis: Since the modified version of CFS signature scheme is secure in the random oracle model [10], We can prove the security of our scheme. The details of our analysis will appear in a full version of the this paper, but we can give some arguments about the security of our scheme. Our scheme satisfies the non-repudiation and the non-forgeability. Indeed, when $N = 1$, our signature scheme degenerates into mCFS signature scheme which satisfies these two properties. If $N > 1$, an attacker who does not belong to the signer set, can not forge the multisignature because

Algorithm 4 CFS-based multisignature

Key Generation: Each signer S_i has to:
- generate his public/private key as in the CFS algorithm,i.e, $H_i = Q_i\widetilde{H}_iP_i$ $(Q_i, \widetilde{H}_i, P_i, \gamma_i)$

Signature:
1- Generation of a random vector $r \in \mathbb{F}_q^{n-k}$
* S_1 selects randomly $k_1 \in \mathbb{F}_q^n$ of weight up to t and computes $r_1 = H_1 \cdot k_1^T$.
* From $i = 2$ to N do
  S_{i-1} broadcasts r_{i-1} to S_i.
  S_i selects randomly $k_i \in \mathbb{F}_q^n$ of weight up to t and computes $r_i = r_{i-1} + H_i \cdot k_i^T$.
* Set $r = r_N$.

2- Multisignature Generation
* S_1 computes a n-bit vector s_1 of weight up to t and an index i_1 s.t. $H_1 \cdot s_1^T = h((M + r)|i_1)$
* For $i := 2$ to N do
  - S_{j-1} sends (s_{j-1}, i_{j-1}) to S_j.
  - S_j checks the validity of s_{j-1} by $H_{j-1} \cdot s_{j-1}^T = h((M+r)|i_{j-1})$ and $\mathsf{w}(s_{j-1}) \leq t$
  - S_j computes a n-bit vector s_j of weight up to t and an index i_j s.t. $H_j \cdot s_j^T = h((H_{j-1} \cdot s_{j-1}^T + h(M + r))|i_j)$
* Set $s = s_N$.
* $\sigma_{cfs} = (s, i_1, \ldots, i_N, r)$ is the multisignature.

Verification: Given a tuple $\sigma = (s, i_1, \ldots, i_N, r)$
* Check that $\mathsf{w}(s) \leq t$
* Compute $x = H_N \cdot s^T$.
* Compute iteratively the sequence $(z_i)_{i=1,\ldots,N}$ defined by:
  - $z_1 = h((M + r)|i_1)$
  - For $j := 2$ to $N : z_j = h((z_{j-1}+h(M+r))|i_j)$.
* The multisignature σ is valid if x and z_N are equals.

the signer set has been already known in advance and if he generates a couple (s_A, i_A) as own signature, this signature will be invalid after checking it by the next signer.

C. KKS-based serial multisignature scheme

1) Description: Our scheme extends the regular KKS-signature into a multi-signer one. In this scheme each signer applies the KKS-signature algorithm to produce his own signature on received components before he forwards it to the next signer for consecutive handling. Before the beginning of signing process, all signers first collaborate to create a public a vector r of $\{0,1\}^{n-k}$ in a serial way which will be concatenated with the original message M. During the signing step, a signer S_i has first to verify the previous signature σ_{i-1} generated by previous signer and then to produce his own signature σ_i as follows: The signer hashes the bitwise addition of M linked with r and the preceding KKS-signature σ_{i-1} generated by S_{i-1} and then he applies the KKS-algorithm on the result. After that, he replaces the resulting signature by substraying the quantity $(\sigma_{i-1} \cdot G_i)$ from it. The last operation is designed in order that the new signature can be verified by the succeeding signer in the same manner as the standard KKS-signature. The KKS-multisignature σ_{kks} consists finally of the KKS-signature produced by the last signer in the queue (say s_N) and the vector r constructed before. To test whether this multisignature is valid, the verifier has to apply the KKS-verification algorithm. The Algorithm 5 explains in more detail our scheme.

Algorithm 5 KKS-based multisignature

Key Generation: Given a hash-function of range $\{0,1\}^{n-k}$, each signer S_i has to:
- select n, k, t_1 and t_2 as security parameter.
- select a random matrix H_i as a parity check matrix of a random (n, k) code $\mathcal{C}_i$.
- Choose secretly and randomly:
  * a generator matrix G_i of a linear code $\mathcal{U}_i$ of length $n' \leq n$ and dimension k s.t. $t_1 \leq \mathsf{w}(u) \leq t_2$ for all $u \in \mathcal{U}_i$.
  * a subset J_i of $\{1, \ldots, n\}$ of cardinality n'.
- Build the sub matrix $H_i(J_i)$ of H_i consisting of the columns h_j where $j \in J_i$.
- Define the matrix $F_i = H_i(J_i)G_i^T$
- The public key: (F_i, H_i, t_1, t_2)
- The private key : (J_i, G_i).

Signature:
1- Generation of a random vector $r \in \mathbb{F}_q^{n-k}$
* S_1 selects a random vector $r_1 \in \mathbb{F}_q^{n-k}$.
* For $i := 2$ to N do
  - S_{i-1} broadcasts r_{i-1} to S_i.
  - S_i selects a random vector $r_i \in \mathbb{F}_q^{n-k}$ and assigns $(r_{i-1}+r_i)$ to r_i, i.e. $r_i \leftarrow (r_{i-1}+r_i)$.
* Set $r = r_N$.

2- Multisignature Generation
* S_1 calculates $\sigma_1^* = h(M|r) \cdot G_1$ and produces σ_1 s.t.
$$\sigma_{1,j} = \begin{cases} \sigma_{1,j}^* & \text{if } j \in J_1, \\ 0 & \text{if } j \notin J_1. \end{cases}$$
* For $i := 2$ to N do
  - S_{i-1} sends σ_{i-1} to S_i.
  - S_i checks the validity of σ_{i-1} by $t_1 \leq \mathsf{w}(\sigma_{i-1}) \leq t_2$ and $F_{i-1} \cdot (h(M|r))^T = H_{i-1} \cdot \sigma_{i-1}^T$.
  - S_i calculates $\sigma_i^* = (h(M|r) + \sigma_{i-1}) \cdot G_i$ and produces σ_i s.t.
$$\sigma_{i,j} = \begin{cases} \sigma_{i,j}^* & \text{if } j \in J_i, \\ 0 & \text{if } j \notin J_i. \end{cases}$$
  - S_i replaces σ_j^* by the quantity $(\sigma_i^* - \sigma_{i-1} \cdot G_i)$
* The multisignature is $\sigma_{kks} = (\sigma_N, r)$.

Verification: Given a tuple (z, r), the multisignature is valid if:
* $t_1 \leq \mathsf{w}(z) \leq t_2$
* $F_N \cdot (h(M|r))^T = H_N \cdot z^T$.

2) Performance Analysis: In [17], three KKS-signature schemes were proposed, named KKS-1, KKS-2 and KKS-3 in [8] respectively. The KKS-1 version introduced an equidistant code $(t = t_1 = t_2)$ of length $n' = 2^k - 1$ correcting $t = 2^k$ errors, where k is its dimension. However, since the length

of this code is huge for any practical applications, the KKS-1 is still impracticable. Therefore, [17] replaced the equidistant code by another code whose non-zero codewords have a weight between two different values t_1 and t_2 and proposed two improvements of KKS-1, KKS-2 and KKS-3.
The KKS-2 is based on the dual of a BCH code while the KKS-3 is fully random construction and uses a random linear code. In this section we restrict our analysis to the KKS-3 signature scheme.

In KKS-3, each signer choose a random $k \times n'$ generator matrix G_i given in the systematic form $[I_k|B_i]$. The public key is composed of F_i and $H_i = [I_r|D_i]$ where D_i is a random $r \times (n-k)$ binary matrix. The secret key consists of the set J_i and the matrix B_i. Thus, to store each public key, we need in total $r(n-r+k)$ bits. For each secret key, we have to store $nh_2(\frac{n'}{n})$ + $k(n'-k)$ bits[1], where $h_2(x) = -x\log_2(x) - (1-x)\log_2(1-x)$. The multisignature consists of a vector of length n and a weight up to t_2 and a random vector of $\{0,1\}^{n-k}$. Thus, the total length of our multisignature is about $\lceil t_2 h_2(\frac{t_2}{n})\rceil + (n-k)$ bits which not depends on the number of signers. The essential part in generating the multisignature is the second step in which each user has to produce his own KKS-like signature while the first phase for producing a common random vector can be performed off-line. Thus, to generate a multisignature, each signer first have to verify the preceding signature and then to produce his KKS-signature. Therefore, the overall cost of our multisignature is approximate to $Nn'k + (N-1)r(n+k)$ binary operations. After receiving a multisignature, any user can check its validity by comparing the results of two vector-matrix multiplications that require about $r(n+k)$ binary operations.

3) Security Analysis: In [17] the authors claimed that their constructions are secure as Niederreiter scheme if the public parameters do not provide any information. Unfortunately [8] showed that a generated KKS-signature discloses a lot of information about the secret set J leading to find the secret matrix G with high probability. Furthermore, [8] proved that just a few intercepted signatures damages the KKS-system. For instance, an attacker needs at most 20 signatures to break the original KKS-3 scheme with an approximate amount of 2^{77} binary operations. Regarding the security of our multisignature, since our construction is based on the KKS-signature, we can assume that our scheme is a few times. Following [8], we propose the same parameters for our multisignature scheme to achieve a security level more than 2^{80}. These parameters are as follows: $n = 2000$, $k = 160$, $n' = 1000$, $r = 1100$, $t_1 = 90$ and $t_2 = 110$.

V. Conclusion

We have proposed two multisignature schemes using error correcting codes that are the first non-generic constructions in post-quantum cryptography. Our schemes make use of the CFS signature KKS-signature scheme and achieve signatures of size $377 + 18.47N$ bits and 1873.8 bits, respectively, both for a security of more than 2^{80} binary operations. However, the first system suffers from slow signature cost and large key sizes while the second scheme is only few times and very fast but also requires big key sizes. Recently, two works are published for reducing the key sizes (see [3], [24]) and further progress on this topic should increase significantly the performance of our schemes.

[1]We use the approximation $\binom{a}{b} \approx 2^{ah_2(\frac{b}{a})}$

References

[1] S. Barg. Some New NP-Complete Coding Problems. *Probl. Peredachi Inf.*, 30:23–28, 1994.

[2] M. Bellare and G. Neven. Multi-signatures in the plain public-key model and a general forking lemma. In *CCS '06: Proc. of the 13th ACM conference on Computer and communications security*, pages 390–399. ACM, 2006.

[3] T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani. Reducing key length of the McEliece cryptosystem. In *Progress in Cryptology – Africacrypt'2009*, LNCS, pages 77–97. Springer, 2009.

[4] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[5] D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. Cryptology ePrint Archive, Report 2008/318, 2008. http://eprint.iacr.org/.

[6] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO '01: Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pages 213–229. Springer, 2001.

[7] D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, pages 416–432. Springer, 2003.

[8] P.L. Cayrel, A. Otmani, and D. Vergnaud. On Kabatianskii-Krouk-Smeets Signatures. In *Proceedings of the first International Workshop on the Arithmetic of Finite Fields (WAIFI 2007)*, Springer, pages 237–251, Madrid, Spain, June 21–22 2007.

[9] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology – Asiacrypt'2001*, volume 2248 of *LNCS*, pages 157–174, Gold Coast, Australia, 2001. Springer.

[10] L. Dallot. Towards a concrete security proof of courtois, finiasz and sendrier signature scheme. Proceedings of WEWoRC 2007, Bochum, Germany,, 2007. http://users.info.unicaen.fr/~ldallot/download/articles/CFSProof-dallot.pdf.

[11] M. Finiasz. *Nouvelles constructions utilisant des codes correcteurs d'erreurs en cryptographie à clef publique*. PhD thesis, INRIA-Ecole polytechnique, October 2004.

[12] M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In *to appear in Advances in Cryptology – Asiacrypt'2009*, 2009. http://eprint.iacr.org/2009/414.pdf.

[13] T. Hardjono and Y. Zheng. A practical digital multisignature scheme based on discrete logarithms (extended abstract). In *in AUSCRYPT'92*, pages 122–132. Springer, 1993.

[14] L. Harn and T. Kiesler. Rsa blocking and multisignature schemes with no bit expansion. *Electron Letters*, 26(18):1490.1491, August 1990.

[15] L. Harn and T. Kiesler. New scheme for digital multisignature. *Electron Letters*, 25(15):1002.1003, July 1989.

[16] K. Itakura and K. Nakamura. New scheme for digital multisignature. *NEC Research and Development*, 71:1–8, October 1983.

[17] G. Kabatianskii, E.Krouk, and B. J. M. Smeets. A digital signature scheme based on random error-correcting codes. *IMA Int. Conf.*, Springer LNCS 1355:161–167, 1997.

[18] K. Kawauchi and M. Tada. On the exact security of multi-signature schemes based on rsa. In *ACISP 2003*, volume 2727.

[19] K. Kawauchi and M. Tada. On the security and the efficiency of multi-signature schemes based on a trapdoor one-way permutation. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E88-A(5):1274–1282, 2005.

[20] Y. Komano, K. Ohta, A. Shimbo, and S. Kawamura. Formal security model of multisignatures. In *ISC*, pages 146–160, 2006.

[21] E. Liberty and S. W. Zucker. The mailman algorithm: A note on matrix-vector multiplication. *Inf. Process. Lett.*, 109(3):179–182, 2009.

[22] R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. Jpl dsn progress report 42-44 , pages 114-116, 1978.

[23] S. Micali, K. Ohta, and L. Reyzin. Accountable-subgroup multisignatures: extended abstract. In *ACM Conference on Computer and Communications Security*, pages 245–254, 2001.

[24] R. Misoczki and P. S. L. M. Barreto. Compact mceliece keys from goppa codes. Preprint, 2009. http://eprint.iacr.org/2009/187.pdf.

[25] S. Mitomi and A. Miyaji. A general model of multisignature schemes with message flexibility, order flexibility, and order verifiability. *IEICE Trans. Fundam.*, E-84-A(5):2488–2499, 2001.

[26] T. Okamoto. A digital multisignature scheme using bijective public-key cryptosystems. *ACM Trans. Comput. Syst.*, 6(4):432–441, 1988.

[27] O.Kazuo and O. Tatsuaki. A digital multisignature scheme based on the fiat-shamir scheme. In *ASIACRYPT '91: Proc. of the International Conference on the Theory and Applications of Cryptology*, pages 139–148. Springer, 1993.

[28] A. Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53. Springer-Verlag., 1984.

[29] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.

[30] L. Wang, E. Okamoto, Y. Miao, T. Okamoto, and H. Doi. Id-based series-parallel multisignature schemes for multi-messages from bilinear maps. In *WCC*, pages 291–303, 2005.