Publ. Math. Debrecen
Supplementum **100** (2022), 567–582
DOI: 10.5486/PMD.2022.Suppl.2

Using GeMSS128 in a ring signature scheme

By Viliam Hromada and Daniela Leščinská

Abstract. In this paper, we investigate the possibility of using the multivariate GeMSS signature scheme as the building block of a ring signature scheme. Namely, we determine a set of parameters which provide 128-bit level of security for different numbers of ring's participants. We also provide performance measurements and compare this ring signature scheme with a similar one which uses the Rainbow signature scheme as its building block instead.

1. Introduction

Currently, nearly all used public-key cryptographic primitives are based either on the problem of integer factorization or the discrete-logarithm problem. Both of these problems could be solved in polynomial time on a quantum computer using the Shor's algorithm [16]. Therefore, as an answer to the need for new quantum-secure cryptographic algorithms, the Post-Quantum Cryptography Standardization was launched by NIST in 2016 with the goal of identifying suitable candidates for quantum-secure public-key cryptographic algorithms. These algorithms would have to be based on mathematical problems not affected by quantum computers.

One of these problems is the problem of solving a system of multivariate non-linear equations over a finite field. Cryptographic algorithms, whose security is built on this problem, are termed multivariate cryptography [6]. This area provides promising candidates, especially in the field of digital signatures. The

Mathematics Subject Classification: 94A60, 94A62.
Key words and phrases: post-quantum cryptography, ring signature scheme, GeMSS.
This paper was sponsored by the NATO Science for Peace and Security Programme under Grant G5448.

third round of the NIST NQC project included two multivariate digital signature algorithms: Rainbow [7], and as an alternative candidate GeMSS [4]. Both are standard digital signature algorithms, i.e., one entity generates a signature and another may verify it. However, more advanced signature schemes are also needed in practice, e.g., a ring signature scheme.

A ring signature scheme is a signature scheme where a user can sign messages anonymously as a member of some group $\mathcal{R}$. The verifier can verify, whether a signature was generated by a member of the group $\mathcal{R}$, but cannot reveal the identity of the signer. In 2017, Mohamed and Petzoldt proposed an efficient multivariate ring signature scheme [11] with a simple design that allows the participants to use an arbitrary standard multivariate signature scheme as a building block.

In this ring signature scheme, each member of the group $\mathcal{R}$ generates an instance of a private and a public key of some multivariate signature scheme. The original proposal of [11] uses the multivariate signature scheme Rainbow. In our work, we investigate the possibility of using GeMSS as the main building block. A similar work has been already done by Demircioglu et. al. in [5]. However, the mentioned paper only proposes the usage of GeMSS and omits any concrete performance results or a proposal of parameter values to achieve desired levels of security. We propose parameters for using GeMSS in a ring signature scheme with 128 bits of security and we also give the performance results.

The structure of this paper is as follows. In Section 2, we summarize the ring signature scheme presented in [11] and GeMSS signature scheme [4]. In Section 3, we determine a set of parameters for GeMSS128 to achieve 128 bits of security for a different number of participants. In Section 4 we present the resulting performance of the scheme and we conclude the paper in Section 5.

2. Preliminaries

In multivariate public-key cryptosystems, the public key is usually a system of m quadratic multivariate polynomials of n indeterminates $(x_1, ..., x_n)$ over a finite field $\mathbb{F}_q$ of q elements, i.e.,

$$\begin{aligned} p^{(1)}(x_1, x_2, ..., x_n) &= \sum_{1 \le i \le j \le n} a_{ij}^{(1)} x_i x_j + \sum_{1 \le i \le n} b_i^{(1)} x_i + c^{(1)}, \\ &\vdots \\ p^{(m)}(x_1, x_2, ..., x_n) &= \sum_{1 \le i \le j \le n} a_{ij}^{(m)} x_i x_j + \sum_{1 \le i \le n} b^{(m)} x_i + c^{(m)}, \end{aligned}$$

where $a_{ij}^{(k)}, b_i^{(k)}, c^{(k)} \in \mathbb{F}_q$, $1 \leq k \leq m$. The problem of solving such a quadratic system is called the MQ-problem and is known to be NP-hard [9]. In order to construct a digital signature based on the MQ-problem, one starts with an easily-invertible polynomial system $\mathbf{f} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, also called the central map. Next, the structure of the central map $\mathbf{f}$, which allows its inversion, is hidden by two invertible affine maps $\mathcal{T} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $\mathcal{S} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ to produce the resulting public key in the following way:

$$\mathcal{P} = \mathcal{T} \circ \mathbf{f} \circ \mathcal{S} : \mathbb{F}_q^n \to \mathbb{F}_q^m.$$

The public key is then the system $\mathcal{P}$ of m polynomials of n indeterminates, the triplet $(\mathcal{T}, \mathbf{f}, \mathcal{S})$ forms the private key.

2.1. GeMSS Signature Scheme. GeMSS [4] is a multivariate signature algorithm chosen as an alternative candidate in the third round of the NIST PQC project.

Key generation. In order to generate a private key, one first generates a random polynomial F of degree D over some n-th degree extension $\mathbb{F}_{2^n}$ of the finite field $\mathbb{F}_2$. The polynomial F is of the form:

$$F(X, v_1, ..., v_v) = \sum_{\substack{0 \leq j < i < n \\ 2^i + 2^j \leq D}} A_{i,j} X^{2^i + 2^j} + \sum_{\substack{0 \leq i < n \\ 2^i \leq D}} \beta_i(v_1, ..., v_v) X^{2^i} + \gamma(v_1, ..., v_v),$$

where the indeterminates $(X, v_1, ..., v_v) \in \mathbb{F}_{2^n} \times \mathbb{F}_2 \times ... \times \mathbb{F}_2$, the coefficients $A_{i,j} \in \mathbb{F}_{2^n}$, the mappings $\beta_i : \mathbb{F}_2^v \to \mathbb{F}_{2^n}$ are linear and the mapping $\gamma : \mathbb{F}_2^v \to \mathbb{F}_{2^n}$ is quadratic. The indeterminates (variables) $(v_1, ..., v_v)$ are called the *vinegar indeterminates (variables)*. After assigning values to these vinegar indeterminates, we obtain a polynomial of the form

$$F_{\mathbf{v}}(X) = \sum_{\substack{0 \leq j < i < n \\ 2^i + 2^j \leq D}} A'_{i,j} X^{2^i + 2^j} + \sum_{\substack{0 \leq i < n \\ 2^i \leq D}} B'_i X^{2^i} + C',$$

which is a univariate polynomial over the field $\mathbb{F}_{2^n}$, $A'_{i,j}, B'_i, C' \in \mathbb{F}_{2^n}$ and whose roots can be found by Berlekamp's algorithm [2] or by the Cantor–Zassenhaus algorithm [3], provided that the degree D is sufficiently small.

Further, let ϕ be an isomorphism $\phi : \mathbb{F}_2^n \to \mathbb{F}_{2^n}$ and let $\psi = \phi \times \mathrm{id}_v : \mathbb{F}_2^{n+v} \to \mathbb{F}_{2^n} \times \mathbb{F}_2^v$, such that if $v = 0$, then $\phi = \psi$. Let $\mathbf{f}$ be the composition

$\mathbf{f} = \phi^{-1} \circ F \circ \psi$, then $\mathbf{f}$ is a system of n quadratic polynomials of $n+v$ indeterminates over $\mathbb{F}_2$, $\mathbf{f} : \mathbb{F}_2^{n+v} \to \mathbb{F}_2^n$.

To hide the structure of the quadratic mapping $\mathbf{f}$, GeMSS uses two random invertible linear mappings $(\mathcal{S}, \mathcal{T}) \in GL_{n+v}(\mathbb{F}_2) \times GL_n(\mathbb{F}_2)$. The public key is then the composition $\mathcal{P} = \mathcal{T} \circ \mathbf{f} \circ \mathcal{S}$ with last Δ polynomials removed: $\mathcal{P} : (\mathbb{F}_2)^{n+v} \to (\mathbb{F}_2)^{n-\Delta}$:

$$\begin{aligned} p^{(1)}(x_1, x_2, ..., x_{n+v}) &= \sum_{1 \le i < j \le n+v} a_{ij}^{(1)} x_i x_j + \sum_{1 \le i \le n+v} b_i^{(1)} x_i + c^{(1)}, \\ &\vdots \\ p^{(n-\Delta)}(x_1, x_2, ..., x_{n+v}) &= \sum_{1 \le i < j \le n+v} a_{ij}^{(n-\Delta)} x_i x_j + \sum_{1 \le i \le n+v} b^{(n-\Delta)} x_i + c^{(n-\Delta)}, \end{aligned}$$

where $a_{ij}^{(k)}, b_i^{(k)}, c^{(k)} \in \mathbb{F}_2$, $1 \le k \le n - \Delta$.

Signature Generation. Suppose $d \in \{0,1\}^*$ is a document to be signed.

(1) Let $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_2^{n-\Delta}$ be a hash function with output length $n - \Delta$. Use it to compute $\mathbf{w} = \mathcal{H}(d)$, i.e., the hash $\mathbf{w}$ of the document d.

(2) Generate Δ random values $\mathbf{r} = (r_1, ..., r_\Delta) \in \mathbb{F}_2^\Delta$ and append them to $\mathbf{w}$ to create $\mathbf{w'} = (\mathbf{w}, r_1, ..., r_\Delta) \in \mathbb{F}_2^n$.

(3) Compute $\mathbf{y} = \mathcal{T}^{-1}(\mathbf{w'})$ and lift the binary vector $\mathbf{y}$ to the extension field $\mathbb{F}_{2^n}$, i.e., $Y = \phi(\mathbf{y})$.

(4) Randomly choose $\mathbf{v} = (v_1, ..., v_v) \in \mathbb{F}_2^v$ and substitute them into the vinegar variables of F to obtain a univariate polynomial $F_\mathbf{v}$ of the form (2), i.e., $F_\mathbf{v}(X) = F(X, \mathbf{v})$.

(5) Find a root Z of the polynomial $F_\mathbf{v}(X) - Y = 0$, e.g. by Berlekamp's algorithm. If such a root does not exist, go back to step (2).

(6) Set $\mathbf{z} = \psi^{-1}(Z, \mathbf{v}) \in \mathbb{F}_2^{n+v}$.

(7) Set $\mathbf{s} = \mathcal{S}^{-1}(\mathbf{z})$. Then $\mathbf{s} \in \mathbb{F}_2^{n+v}$ is the corresponding digital signature of d.

Signature Verification. Suppose $d \in \{0,1\}^*$ is a document and $\mathbf{s} \in \mathbb{F}_2^{n+v}$ its corresponding signature.

(1) Let $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_2^{n-\Delta}$ be a hash function with output length $n - \Delta$. Use it to compute $\mathbf{w} = \mathcal{H}(d)$, i.e., the hash $\mathbf{w}$ of the document d.

(2) Check if $\mathbf{w} \stackrel{?}{=} \mathcal{P}(\mathbf{s})$. If yes, the signature has been verified. Otherwise, the signature has been rejected.

The relation between the signature **s** and the signed hash **w**, can be visualized as follows:

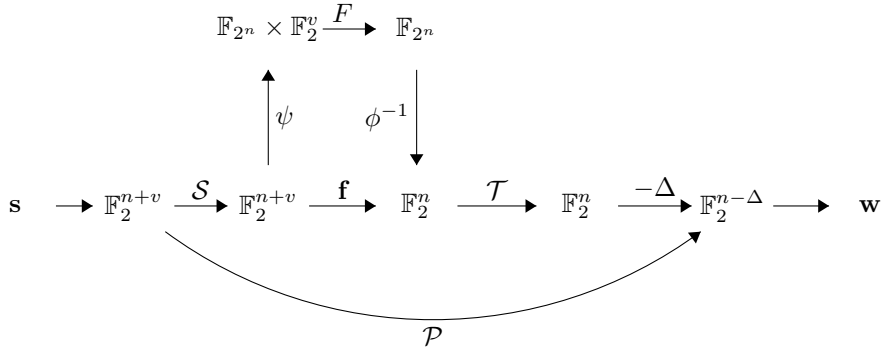

Figure 1. The relation between the signature **s** and the signed hash **w**

2.2. Ring Signature Scheme. Ring signature schemes were first proposed by Rivest et. al. in 2001 [14]. A ring signature scheme is a signature scheme which allows a signer to sign messages anonymously on behalf of some group of members $\mathcal{R}$. The verifier can verify, whether the signature was generated by a member of the group $\mathcal{R}$, but cannot reveal the identity of the signer, nor there exists any group manager, who is able to reveal the identity of the signer (as opposed to the group signatures).

A number of multivariate ring signature schemes have been proposed in recent years, e.g. the scheme by Wang [19] or by Wang et. al. [20]. The former is based on the multivariate identification scheme by Sakumoto et. al [15], but unfortunately, enables the adversary to forge a signature with probability $\frac{2}{3}$. The latter scheme enables to construct a ring signature scheme from a standard multivariate signature scheme. Due to this design, the security of the scheme is based on the security of the underlying multivariate signature.

In 2017, a multivariate ring signature scheme was proposed by Mohamed and Petzoldt in [11], which similarly to [20] allows the participants to use an arbitrary standard multivariate signature scheme as a building block, but has a simpler design and faster signature generation and verification. Next, we describe this scheme, as presented in [11].

Let $\mathcal{R} = \{u_1, u_2, ..., u_k\}$ be a ring of users.

Key generation. Each user u_i generates a key-pair $((\mathcal{T}_i, \mathbf{f}_i, \mathcal{S}_i), \mathcal{P}_i)$ of some underlying standard multivariate digital signature scheme. The public key of the ring

is the concatenation of all individual public keys, i.e., $\mathcal{P} = \mathcal{P}_1||\mathcal{P}_2||...||\mathcal{P}_k$, while each user u_i keeps $(\mathcal{T}_i, \mathbf{f}_i, \mathcal{S}_i)$ as his private key sk_i. Generally, let each public key $\mathcal{P}_i$ be a system of m polynomials in n indeterminates over a finite field $\mathbb{F}$, $\mathcal{P}_i : \mathbb{F}^n \to \mathbb{F}^m$.

Ring signature generation. In order to sign a message d on behalf of the ring $\mathcal{R}$, the user u_i uses a hash function $\mathcal{H}$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ of the message. Then he chooses $k-1$ random vectors $\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, ..., \mathbf{z}_k \in \mathbb{F}^n$ and computes

$$\tilde{\mathbf{w}} = \mathbf{w} - \sum_{\substack{j=1 \\ j \neq i}}^{k} \mathcal{P}_j(\mathbf{z}_j) \in \mathbb{F}^m$$

and uses his private key to compute a vector $\mathbf{z}_i \in \mathbb{F}^n$ such that $\mathcal{P}_i(\mathbf{z}_i) = \tilde{\mathbf{w}}$. The ring signature of the message d is then $(\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k) \in \mathbb{F}^{kn}$.

Ring signature verification. In order to verify if $(\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k) \in \mathbb{F}^{kn}$ is a signature of the message d, the verifier computes the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ of the message d and uses the public keys $\mathcal{P}_1, ..., \mathcal{P}_k$ to compute

$$\hat{\mathbf{w}} = \sum_{j=1}^{k} \mathcal{P}_j(\mathbf{z}_j) \in \mathbb{F}^m.$$

If $\hat{\mathbf{w}} = \mathbf{w}$ holds, then the signature is accepted, otherwise it is rejected.

Security of the ring signature scheme. The basic security criteria of a ring signature scheme are anonymity and unforgeability:

- Anonymity: The receiver of the signed message should not be able to reveal the actual identity of the signer.
- Unforgeability: Given a message d, an adversary $\mathcal{A}$ not belonging to the ring $\mathcal{R}$ of legitimate signers is not able to forge a valid ring signature for the message d on behalf of the ring $\mathcal{R}$.

The authors prove in their paper [11] that this ring signature scheme provides perfect anonymity, i.e., the ring signature contains no information, which member of the ring generated the signature and even a computationally unrestricted adversary can not reveal the identity of the signer.

In order to forge a signature of a message with hash $\mathbf{w}$ on behalf of a ring $\mathcal{R} = \{u_1, ..., u_k\}$ of signers, the attacker has to find a solution $(\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_k)$ of the equation

$$\mathcal{P}_1(\mathbf{z}_1) + \mathcal{P}_2(\mathbf{z}_2) + ...\mathcal{P}_k(\mathbf{z}_k) = \mathbf{w}.$$

There are two approaches to do it:

(1) The first approach is to randomly generate $(\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{k-1})$, compute $\tilde{\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{k-1} \mathcal{P}_i(\mathbf{z}_i)$ and try to find a solution of $\mathcal{P}_k(\mathbf{z}_k) = \tilde{\mathbf{w}}$. This is an equivalent of breaking the underlying standard multivariate signature scheme.

(2) The second approach is to solve the system (3) directly as an undetermined system of multivariate equations.

The second approach is not as difficult as the first one, i.e., breaking the underlying scheme. This is due to the fact that the system (2.2) is a highly undetermined system – and the higher the number k of participants in the ring, the higher the number of variables in the system (2.2). For systems of equations of this type, there are two important results we have to consider [11]:

(1) If the number of variables n in an undetermined multivariate quadratic system $\mathcal{P}$ of m equations is given by $n = \omega \cdot m$, then a solution of the system $\mathcal{P}$ can be found in the same time as finding a solution of a determined system of $m - \lfloor \omega \rfloor + 1$ equations [18].

(2) If the number of variables n in the multivariate quadratic system $\mathcal{P}$ of m equations exceeds $n \geq \frac{m(m+3)}{2}$, $\mathcal{P}$ can be solved in polynomial time [10].

3. GeMSS ring signature scheme parameters

Originally, the authors of [11] instantiate the ring signature scheme with the signature scheme Rainbow as its building block and propose a set of parameters to achieve 80, 100 and 128-bit level of security for rings of 5, 10, 20 and 50 users. In this paper, we propose a set of parameters for GeMSS to be used in this ring signature scheme to achieve 128-bit level of security for rings of 5, 10, 20 and 50 users. The parameters have to fulfil two conditions:

(1) Each instance of GeMSS must be secured at the desired level of security, i.e., 128 bits.

(2) The resulting system of m equations of nk variables (3) must have a complexity of solving as at least a determined system of $m - \lfloor \frac{nk}{m} \rfloor + 1$ equations, i.e., 128 bits.

3.1. Security of one instance of GeMSS. GeMSS has four parameters, which influence its security:

- n, the degree of the field extension of $\mathbb{F}_2$;
- D, the degree of the private-key polynomial F;

- Δ, the number of removed polynomials during the public-key generation;
- v, the number of vinegar variables in the private-key polynomial F.

In the 3rd round of NIST PQC project, there are three main sets of parameters for GeMSS proposed [4], offering 128, 192 and 256 bits of security. They are summarized in Table 1.

Version	(n, D, Δ, v)	polynomials $(n - \Delta)$	indeterminates $(n + v)$
GeMSS128	$(174, 513, 12, 12)$	162	186
GeMSS192	$(265, 513, 22, 20)$	243	285
GeMSS256	$(354, 513, 30, 33)$	324	387

Table 1. GeMSS parameters as proposed in [4]

These parameters have been chosen so that each known attack, either directly aimed at solving a system of quadratic multivariate equations over $\mathbb{F}_2$, or aimed at finding the private-key, should have a complexity of at least $2^{128}, 2^{192}$ and 2^{256}, respectively. The supporting documentation of GeMSS [4] contains a treatment of possible attacks (at the time of publication) and their corresponding complexities, which have to be taken into account when determining a set of parameters. We will list those that are essential to determine the parameters.

First attack is aimed at solving a system of non-linear boolean equations using the `BooleanSolve` algorithm [1]. For a system of m equations and m variables, the Las-Vegas variant of this algorithm has an expected complexity of

$$\mathcal{O}(2^{0.792 \cdot m})$$

and is used as the reference approach to determine the minimal number of polynomials m used in the public key of GeMSS.

Second attack to consider, which is again aimed at solving a system of non-linear equations, is performed by using the F5 algorithm for finding the Gröbner basis of a non-linear system of equations [8]. The complexity of this approach can be bounded by

$$\mathcal{O}\left(\binom{m}{D_{reg}}^2\right),$$

where D_{reg} is the degree of regularity of the corresponding non-linear system. In the case of GeMSS, the degree of regularity of the public-key can be estimated [13] as

$$D_{reg} \geq \left\lfloor \frac{\lfloor \log_2(D-1) \rfloor + 1 + \Delta + v + 7}{3} \right\rfloor.$$

Relations (3.1), (3.1) show the importance of the degree D of the private-key polynomial F, the number of vinegar variables v and the number of removed polynomials Δ in the security of GeMSS, since they all increase the degree of regularity and the complexity of the attack using the F5 algorithm.

It can be seen that parameters presented in Table 1 follow these rules to achieve the desired level of security. However, in 2021, a new key recovery attack on GeMSS was published by Tao, et. al. [17], which shows that GeMSS is not as secure as claimed. The attack falls into the category of so-called MinRank attacks. Without going into too much detail, the complexity of the attack using the support minors modelling is

$$\mathcal{O}\left((n+v)^2\binom{2d+2}{d}+(n+v)\binom{2d+2}{d}^2\right)^{\omega},$$

where $d = \lfloor \log_2(D) \rfloor$ and ω is the linear algebra constant. Using the value $\omega = 2.81$, the complexity of this attack is 2^{118} for GeMSS128, 2^{120} for GeMSS192 and 2^{121} for GeMSS256, which means that all sets of parameters presented in Table 1 are not as secure as declared. The simplest way to thwart this type of attack is to increase the value of D, which unfortunately increases the signature generation time.

However, also in 2021, a projection modifier of GeMSS was proposed by Øygarden, et. al. [12], which could be useful in protecting GeMSS against the rank attack presented in [17]. The idea of the modifier is to project the signature space $\mathbb{F}_2^{n+v}$ to a subspace of lower dimension, $\mathbb{F}_2^{n+v-p}$, according to a parameter p. Implementationally, this means that the signing algorithm is virtually the same, however, the signature $\mathbf{s}$ is accepted only if its last p coordinates are all zero. That means, that in this case, the whole signature process is repeated on average 2^p times. The positive side of this modification is that the degree D of the secret polynomial F can be kept at the original value as proposed by authors of GeMSS. The complexity of the rank attack on projected GeMSS is estimated to be [12]:

$$\mathcal{O}\left((n+v)^2\binom{n'}{d+p}+(n+v)\binom{n'}{d+p}^2\right)^{\omega},$$

where $d = \lceil \log_2(D) \rceil$, $n' = \left\lceil \frac{(n+v)(d+p+1)}{n-\Delta} \right\rceil + d + p + 1$ and ω is the linear algebra constant.

3.2. Security of the ring signature scheme using GeMSS. As mentioned earlier, in order to forge a signature, the attacker does not have to break the

underlying GeMSS instance, he can try to solve the system (2.2) directly. If the GeMSS instance used leads to a public key of m polynomials and $n + v$ indeterminates ($n + v - p$ indeterminates in the case of projected GeMSS), then to forge a signature in a ring of k users, the attacker would have to solve a system of m equations with $k(n + v)$ variables. Due to the result of [18], solving this underdetermined system has the same complexity as solving a system of $m - \lfloor \frac{k(n+v)}{m} \rfloor + 1$ equations.

The best approach to solve such a system is to use the `BooleanSolve` algorithm, which in the case of unprojected GeMSS would have a complexity of

$$\mathcal{O}(2^{0.792 \cdot (m - \lfloor \frac{k(n+v)}{m} \rfloor + 1))})$$

and in the case of projected GeMSS would have a complexity of

$$\mathcal{O}(2^{0.792 \cdot (m - \lfloor \frac{k(n+v-p)}{m} \rfloor + 1))}).$$

We see that the higher the number of participants k, the greater the difference between the number of variables $k(n + v)$ or $k(n + v - p)$ and the number of equations m, which negatively influences the security. Therefore, the parameters have to scale with the number of participants.

3.3. Parameter estimation. Aggregating the results and complexities of different attacks presented in the preceding two subsections, we have chosen the following sets of parameters to construct a ring signature scheme, which uses GeMSS128 as its building block and has 128 bits of security. We propose 4 different sets of parameters according to the maximum number of users that may form a ring, $k \in \{5, 10, 20, 50\}$ and still attain the 128-bit security. We also propose two versions of parameters for GeMSS128 - without and with the projection modifier.

In the unprojected version of GeMSS128, we increased value of the degree D of the secret polynomial F to $D = 2^{11} + 1 = 2049$, in order to thwart the rank attack presented in [17]. Also, due to the increasing number of participants in the ring, the original number of polynomials in the public-key $m = 174$ has to be gradually increased to maintain the level of security at 128 bits. To do this, we increased the parameter n, which in turn influenced both the number of polynomials and the number of indeterminates in the public key, so that not only each instance of GeMSS128 would have a security of 128 bits, but also the system (2.2) would have a complexity of solving at least 2^{128} for each considered number of ring participants. We were able to decrement the value of parameters v and Δ for 20 and 50 participants and still keep the desired security level. We present

the parameters in Table 2 along with the estimated $\log_2$ complexities of presented attacks.

k	5	10	20	50
D	2049	2049	2049	2049
(n, Δ, v)	$(178, 12, 12)$	$(184, 12, 12)$	$(194, 11, 11)$	$(227, 11, 11)$
$\log_2$ of (3.1)	131	136	144	171
$\log_2$ of (3.1)	138	140	136	143
$\log_2$ of (3.1)	140	140	141	141
$\log_2$ of (3.2)	128	128	128	128

Table 2. Parameters for the unprojected GeMSS128 with attack complexities

Table 3 contains parameters for a ring signature scheme based on GeMSS128 using the projection modifier [12]. In this case, we used the results presented in [12] suggesting that using the projection modifier with value $p = 2$ allows to keep the degree D of the secret polynomial F at $D = 2^9 + 1 = 513$ for 128-bit security.

k	5	10	20	50
D	513	513	513	513
(n, Δ, v, p)	$(178, 12, 12, 2)$	$(184, 12, 12, 2)$	$(194, 11, 11, 2)$	$(227, 11, 11, 2)$
$\log_2$ of (3.1)	131	136	144	171
$\log_2$ of (3.1)	132	133	129	135
$\log_2$ of (3.1)	135	135	136	136
$\log_2$ of (3.2)	128	128	128	129

Table 3. Parameters for the projected GeMSS128 with attack complexities

4. Experiments

To measure the performance of the ring signature scheme built on top of GeMSS128 with parameters presented in the previous section, we have implemented the ring signature scheme in C language. We have implemented all presented variants, e.g. for 5, 10, 20 and 50 participants and compared the versions with/without projection modifier. We have also implemented the ring signature scheme which uses Rainbow as its building block and compared this scheme with GeMSS-based scheme as well. All implementations used in the experiments can be found at `https://uim.fei.stuba.sk/en/pracovnici/viliam-hromada/`.

We have used the optimized implementation of GeMSS128, which is available at the website of the 3rd round of the NIST PQC `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-3-submissions`. We have used the optimized implementation of Rainbow, which is available at the website of the 2nd round of the NIST PQC `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-2-submissions`.

We have measured the signature generation time and the signature verification time for each set of parameters for $k = 5, 10, 20, 50$ members of the ring. Measurements were repeated 100 times and averaged, and were performed on a laptop with the following specification: Intel Core i7-3630QM@2.40GHz CPU, 8GB RAM, Ubuntu 18.04.01, GCC 7.5.0. The compilation was made with the following GCC flags: `-O4 -msse2 -msse3 -msse4.1 -mpclmul -mpopcnt -funroll-loops`.

Ring signature with GeMSS128 without projection. Table 4 summarizes the resulting average times for signature generation, signature verification, along with the sizes of ring public key and the ring signature for ring signature scheme which uses GeMSS128 without projection as its building block.

k	5	10	20	50
D	2049	2049	2049	2049
(n, Δ, v)	(178,12,12)	(184,12,12)	(194,11,11)	(227,11,11)
PK size [KB]	1,883	4,151	9,661	38,397
Sig size [b]	1,320	2,720	5,440	15,200
Sig gen [s]	6.47	7.02	12.34	12.44
Sig ver [ms]	0.8	1.4	3.0	10.0

Table 4. Ring signature using GeMSS128 without projection

Ring signature with GeMSS128 with projection. Table 5 summarizes the resulting average times for signature generation, signature verification, along with the sizes of ring public key and the ring signature for ring signature scheme which uses GeMSS128 with projection as its building block.

Ring signature with Rainbow. Table 6 summarizes the resulting average times for signature generation, signature verification, along with the sizes of ring public key and the ring signature for ring signature scheme which uses Rainbow as its building block. The parameters of Rainbow for different ring sizes were taken from the paper [11].

k	5	10	20	50
D	513	513	513	513
(n, Δ, v, p)	(178,12,12,2)	(184,12,12,2)	(194,11,11,2)	(227,11,11,2)
PK size [KB]	1,843	4,067	9,473	37,755
Sig size [b]	1,300	2,660	5,380	15,100
Sig gen [s]	3.64	4.08	6.22	7.06
Sig ver [ms]	0.8	1.4	3.0	9.4

Table 5. Ring signature using GeMSS128 with projection

Rainbow $\mathbb{F}_{256}$	$k = 5$	$k = 10$	$k = 20$	$k = 50$
(v_1, o_1, o_2)	(36,23,20)	(34,26,23)	(32,33,29)	(30,64,58)
PK size [KB]	680	1,708	5,522	70,180
Sig size [b]	3,160	6,640	15,040	60,800
Sig gen [s]	0.001	0.003	0.010	0.103
Sig ver [ms]	1.2	3.0	9.6	105

Table 6. Ring signature using Rainbow (parameters from [11])

Discussion. As can be seen from the results, if we compare the ring signature scheme built upon GeMSS128 without projection and the ring signature scheme built upon GeMSS128 with projection, clearly the largest difference is the faster signature generation in the projected version. This is of course due to the fact that the degree D of the secret polynomial is $D = 513$ in the projected version and $D = 2049$ in the unprojected version. This degree plays an important part in the complexity of the signature generation, as it influences the complexity of the Berlekamp's factorisation algorithm. Therefore performance-wise, the recommendation is clearly to use the projected version of GeMSS digital signature scheme.

When comparing GeMSS and Rainbow (with parameters taken from [11]), for rings of smaller sizes, e.g. 5, 10 or 20 users, clearly Rainbow-based ring signature scheme offers advantages over GeMSS-based ring signature scheme, due to lower public-key size and much faster signature generation. However, GeMSS128 retains its property of short signatures, since it can be seen that in all scenarios, the size of ring-signature is smaller in the case of GeMSS-based construction, than in the case of Rainbow-based construction.

Situation changes in the case of a large ring of 50 users, in which the size of the public-key of Rainbow-based ring signature scheme is larger than the public-key

of GeMSS-based ring signature scheme (approx. 70 MB vs. 38 MB). Also the size of the ring signature is approximately four times larger in the Rainbow-based ring scheme than in the GeMSS-based scheme. The difference in signature verification is also more prominent in this case, since it takes approximately 10 milliseconds to verify the ring signature in the case of GeMSS-based scheme and 100 milliseconds to verify the ring signature in the case of Rainbow-based scheme. On the other hand, Rainbow still retains its clear advantage in the signature generation time, taking 103 milliseconds to generate the ring signature vs. 7 seconds it takes to generate the ring signature using the GeMSS-based scheme.

5. Conclusions

In this paper, we propose a set of parameters that can be used in the GeMSS128 signature scheme to create a ring signature scheme, which uses it as its building block. The proposed set of parameters offer 128 bits of security for rings up to $5, 10, 20$ and 50 users. Due to the recent developments in the cryptanalysis of this signature scheme, we propose two sets of parameters: one version with the projection modifier and one without it. The measurements show that the projected version offers performance advantage over the unprojected version. We have also compared the GeMSS-based ring signature scheme with the Rainbow-based ring signature scheme, which also offers 128 bits of security. The experiments show that for smaller rings, Rainbow-based scheme offers advantages in signature generation time and public-key sizes. However, in the case of a large ring of 50 users, GeMSS-based ring scheme offers smaller public keys, smaller ring signatures and faster signature verification, even though Rainbow-based scheme still offers clear advantage in the signature generation time. The disadvantage of proposed parameters is clearly the large value of the degree D of the secret polynomial. This might be remedied by proposing a different set of parameters, similar to RedGeMSS or BlueGeMSS [4], where the degree D is lower, but with a larger value of the projection parameter p, which may be an interesting goal in the future.

References

[1] M. Bardet, J.-C. Faugère, B. Salvy and P. J. Spaenlehauer, On the complexity of solving quadratic boolean systems, *Journal of Complexity* **29**, no. 1 (2013), 53–75.

[2] E. R. Berlekamp, Factoring polynomials over finite fields, *Bell System Technical Journal* **46**, no. 8 (1967), 1853—1859.

[3] G. Cantor and H. Zassenhaus, A new algorithm for factoring polynomials over finite fields, *Mathematics of Computation* **36**, no. 154 (1981), 587–592.

[4] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret and J. Ryckeghem, GeMSS: A great multivariate short signature (Round 3 submission), Technical report, *National Institute of Standards and Technology*, 2020, https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[5] M. Demircioglu, S. Akleylek and M. Cenk, Efficient GeMSS based ring signature scheme, *Malaysian Journal of Computing and Applied Mathematics* **3**, no. 1 (2020), 35–39.

[6] J. Ding, A. Petzoldt and D. Schmidt, Multivariate Public Key Cryptosystems, Second Edition, *Springer*, 2020.

[7] J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, B.-Y. Yang, M. Kannwischer and J. Patarin, Rainbow (Round 3 submission), Technical report, *National Institute of Standards and Technology*, 2020, https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions.

[8] J.-C. Faugère and A. Joux, Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases, In: Annual International Cryptology Conference, 2003, 44–60.

[9] M. R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, *W. H. Freeman and Company*, 1979.

[10] H. Miura, Y. Hashimoto and T. Takagi, Extended algorithm for solving underdefined multivariate quadratic equations, In: International Workshop on Post-Quantum Cryptography PQCrypto, Lecture Notes in Computer Science, *Springer*, 2013, 118–135.

[11] M. S. E. Mohamed and A. Petzoldt, RingRainbow – An efficient multivariate ring signature scheme, In: International Conference on Cryptology in Africa, Lecture Notes in Computer Science, *Springer*, 2017, 3–20.

[12] M. Øygarden, D. Smith-Tone and J. Verbel, On the effect of projection on rank attacks in multivariate cryptography, In: International Conference on Post-Quantum Cryptography PQCrypto, Lecture Notes in Computer Science, *Springer*, 2021, 98–113.

[13] A. Petzoldt, On the complexity of the hybrid approach on HFEv-, *IACR Cryptol. ePrint Arch.* (2017), https://eprint.iacr.org/2017/1135.

[14] R. L. Rivest, A. Shamir and Y. Tauman, How to leak a secret, In: International Conference on the Theory and Application of Cryptology and Information Security, Lecture Notes in Computer Science, *Springer*, 2001, 552–565.

[15] K. Sakumoto, T. Shirai and H. Hiwatari, Public-key Identification Schemes Based on Multivariate Quadratic Polynomials, *Advances in Cryptology - CRYPTO 2011*, Lecture Notes in Computer Science, *Springer* (2011), 706–723.

[16] P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Journal on Computing* **26**, no. 5 (1997), 1484–1509.

[17] C. Tao, A. Petzoldt and J. Ding, Efficient key recovery for all HFE signature variants, In: Advances in Cryptology – CRYPTO 2021, Lecture Notes in Computer Science, *Springer*, 2021, 70–93.

[18] E. Thomae and C. Wolf, Solving underdetermined systems of multivariate quadratic equations revisited, In: International Workshop on Public Key Cryptography, Lecture Notes in Computer Science, *Springer*, 2012, 156–171.

[19] L. L. Wang, A new multivariate-based ring signature scheme, *Applied Mechanics and Materials* **347** (2013), 2688–2692.

[20] S. Wang, R. Ma, Y. Zhand and X. Wang, Ring signature scheme based on multivariate public key cryptosystems, *Computers & Mathematics with Applications* **62**, no. 10 (2011), 3973–3979.

VILIAM HROMADA
INSTITUTE OF COMPUTER SCIENCE AND MATHEMATICS
FACULTY OF ELECTRICAL ENGINEERING AND INFORMATION TECHNOLOGY
SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA
ILKOVIČOVA 3, 812 19 BRATISLAVA
SLOVAKIA

E-mail: viliam.hromada@stuba.sk