Efficient Multivariate Ring Signature Schemes

Mohamed Saied Emam Mohamed[1], Albrecht Petzoldt[2]
mohamed@cdc.informatik.tu-darmstadt.de, albrecht.petzoldt@nist.gov

[1] Technische Universität Darmstadt, Germany
[2] National Institute for Standards and Technology, Gaithersburg, Maryland, USA

Abstract. Multivariate Cryptography is one of the main candidates for creating post-quantum cryptosystems. Especially in the area of digital signatures, there exist many practical and secure multivariate schemes. However, there is a lack of more advanced schemes, such as schemes for oblivious transfer and signature schemes with special properties. While, in the last years, a number of multivariate ring signature schemes have been proposed, all of these have weaknesses in terms of security or efficiency. In this paper we propose a simple and efficient technique to extend arbitrary multivariate signature schemes to ring signature schemes and illustrate it using the example of Rainbow. The resulting scheme provides perfect anonymity for the signer (as member of a group), as well as shorter ring signatures than all previously proposed post-quantum ring signature schemes.

Keywords: Multivariate Cryptography, Ring Signatures, Rainbow Signature Scheme

1 Introduction

Cryptographic techniques are an essential tool to guarantee the security of communication in modern society. Today, the security of nearly all of the cryptographic schemes used in practice is based on number theoretic problems such as factoring large integers and solving discrete logarithms. The best known schemes in this area are RSA [22], DSA [13] and ECC. However, schemes like these will become insecure as soon as large enough quantum computers are built. The reason for this is Shor's algorithm [24], which solves number theoretic problems like integer factorization and discrete logarithms in polynomial time on a quantum computer. Therefore, one needs alternatives to those classical public key schemes which are based on hard mathematical problems not affected by quantum computer attacks (so called post-quantum cryptosystems).
Besides lattice, code and hash based cryptosystems, multivariate cryptography is one of the main candidates for this [4]. Multivariate schemes are in general very fast and require only modest computational resources, which makes them attractive for the use on low cost devices like smart cards and RFID chips [5,6]. However, while there exist many practical multivariate standard signature schemes

such as UOV [14], Rainbow [9] and Gui [21], there is a lack of more advanced multivariate schemes such as schemes for oblivious transfer and signature schemes with special properties.
Ring signature schemes allow a user to sign messages anonymously as a member of a group $\mathcal{R}$. The verifier can check, if the message was indeed signed by a member of the group, but has no means to reveal the concrete identity of the signer. Therefore, ring signature schemes are an important tool to secure the privacy of the users. In the last years, a number of multivariate ring signature schemes have been proposed [19,31,28,27]. However, as we find, all of these schemes share certain weaknesses with regard to efficiency or security.
In this paper, we present a new general technique to extend multivariate signature schemes to ring signature schemes. By doing so, we obtain a much simpler construction for multivariate ring signature schemes, which is therefore much easier to understand and analyze than previous constructions. By applying our technique to Rainbow, we obtain a ring signature scheme whose ring signatures are not longer than standard signatures of many other post-quantum signature (e.g. lattice, hash based) schemes. Furthermore, due to the efficiency of the Rainbow scheme, our scheme is very fast.
The rest of this paper is organized as follows. Section 2 reviews the concept of ring signatures and discusses the basic security notions. In Section 3 we give an overview of multivariate cryptography and introduce the Rainbow signature scheme, which is one of the best studied and most efficient multivariate signature schemes. Furthermore, in this section, we consider the existing multivariate ring signature schemes and analyze them with regard to security and performance. Section 4 presents our technique to extend multivariate signature schemes such as Rainbow to ring signature schemes and discusses the security of our construction. In Section 5 we give concrete parameter sets for our scheme based on Rainbow, while Section 6 presents an alternative construction of multivariate ring signatures reducing key and signature sizes. In Section 7 we describe a technique to reduce the public key size further. Section 8 deals with the implementation of our scheme and presents performance results, whereas Section 9 compares our construction with other existing ring signature schemes (both from the classical and the post-quantum world). Finally, Section 10 concludes the paper.

2 Ring Signatures

Ring signature schemes as proposed by Rivest, Shamir and Tauman in [23] allow a signer to sign a message anonymously on behalf of a group $\mathcal{R} = \{u_1, \ldots, u_k\}$ of possible signers. The receiver of a signed message can check, if the message was indeed signed by a member of the group, but can not reveal the concrete identity of the signer. For example, the group of signers could be the set of employees of a company. By verifying the ring signature of a signed document (e.g. a bill), the receiver can ensure that it really was signed by an employee of the given company. By hiding the identity of the actual signer, ring signatures make

therefore an important contribution to secure the privacy of the signer.
The concept of ring signatures is closely related to *group signatures*. However, while, in a group signature scheme, there exists a group manager who can, in the case of a controversy, connect a group signature to the actual signer, such a function does not exist in a ring signature scheme. Therefore, a ring signature scheme provides full anonymity to the signers (as members of the group).
Another related notion is that of *threshold ring signatures*. A threshold ring signature allows a verifier to check if, for any given number $s \in \{1, \ldots, k\}$, at least s members of the group $\mathcal{R}$ contributed to a signature. A basic ring signature scheme is therefore a special case of a threshold ring signature scheme with $s = 1$. Threshold ring signature schemes on the basis of multivariate polynomials have been proposed in [19,31]. However, by restricting to the case of ring signatures, we can reduce the key and signature sizes of the scheme drastically.

Formally, we can define a ring signature scheme $\mathcal{RS}$ as follows [3].
Let $\mathcal{R} = \{u_1, u_2, \ldots, u_k\}$ be a group (called ring) of users. A ring signature scheme consists of the three algorithms `KeyGen`, `RingSign` and `Verify`.

- $\texttt{KeyGen}(1^\lambda)$: The probabilistic algorithm `KeyGen` takes as input a security parameter λ and outputs a key pair (sk, pk). In a ring signature scheme, this algorithm is performed by every user $u_i \in \mathcal{R}$.
- $\texttt{RingSign}(d,\ sk_i,\ \{pk_1, \ldots, pk_k\})$: The (probabilistic) algorithm `RingSign` takes as input the message d to be signed, the secret key sk_i of one user u_i and a list of the public keys $\{pk_1, \ldots, pk_k\}$ of all users $u_j \in \mathcal{R}$. The algorithm outputs a ring signature σ for the message d on behalf of the ring $\mathcal{R}$.
- $\texttt{Verify}((d, \sigma), \{pk_1, \ldots, pk_k\})$: The deterministic algorithm `Verify` takes as input a message/signature pair (d, σ) and a list of public keys $\{pk_1, \ldots, pk_k\}$. It outputs **TRUE**, if σ is a valid ring signature for the message d on behalf of the ring $\mathcal{R}$, and **FALSE** otherwise.

We assume that the ring signature scheme $\mathcal{RS}$ is *correct*, i.e.

$$\texttt{Pr}[\texttt{Verify}((d, \texttt{RingSign}(d, sk_i, \{pk_1, \ldots, pk_k\}), \{pk_1, \ldots, pk_k\}) = 1$$

for all $i \in \{1, \ldots, k\}$.

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

- **Anonymity**: The receiver of a signed message should not be able to detect the concrete identity of the signer. More formally, anonymity can be defined using the following security game.

  Game[Anonymity]:
  1. The algorithm `KeyGen` is used to generate k key pairs $((sk_1, pk_1), \ldots, (sk_k, pk_k))$. The set of public keys $\{pk_1, \ldots, pk_k\}$ is given to the adversary $\mathcal{A}$.

2. The adversary $\mathcal{A}$ is given access to a signing oracle $\mathcal{OS}(i, d)$, which, on input of an index $i \in \{1, \ldots, k\}$ and a message d returns a valid ring signature σ for the message d on behalf of the ring $\mathcal{R} = \{u_1, \ldots, u_k\}$. Hereby, in order to create the signature σ, the signing oracle $\mathcal{OS}$ uses the secret key sk_i of the user u_i.
3. $\mathcal{A}$ outputs a message $d^\star$ as well as two indices i_0 and $i_1 \in \{1, \ldots, k\}$. He is given a signature $\sigma \leftarrow \texttt{RingSign}(d^\star, sk_{i_b}, \{pk_1, \ldots, pk_k\})$, where b is randomly chosen from $\{0, 1\}$.
4. The adversary $\mathcal{A}$ outputs a bit b'. He wins the game, if and only if $b' = b$ holds.

The ring signature scheme$\mathcal{RS}$ is said to provide anonymity, if the advantage

$$\text{Adv}_{\mathcal{A}} = 2 \cdot \Pr[b' = b] - 1$$

is, for every PPT adversary $\mathcal{A}$, negligible.

– **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 σ for the message d on behalf of the ring $\mathcal{R}$.
More formally, we can define unforgeability using the following game

Game[Unforgeability]:

1. The algorithm `KeyGen` is used to generate k key pairs $((sk_1, pk_1), \ldots, (sk_k, pk_k))$. The set of public keys $\{pk_1, \ldots, pk_k\}$ is given to the adversary $\mathcal{A}$.
2. The adversary $\mathcal{A}$ is given access to a signing oracle $\mathcal{OS}(d)$, which, on the input of a message d, returns a valid ring signature σ for the message d on behalf of the ring $\mathcal{R} = \{u_1, \ldots, u_k\}$.
3. $\mathcal{A}$ is given a challenge message $d^\star$. He wins the game, if he is able to produce a valid ring signature $\sigma^\star$ for $d^\star$ on behalf of the ring $\mathcal{R}$.

The ring signature scheme $\mathcal{RS}$ is said to provide unforgeability, if the success probability

$$\Pr_{\mathcal{A}}[\text{success}] = \Pr[\texttt{Verify}((d^\star, \sigma^\star), \{pk_1, \ldots, pk_k\}) = \textbf{TRUE}]$$

is, for any PPT adversary $\mathcal{A}$, negligible.

3 Multivariate Cryptography

The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials (see equation (1)).

$$\begin{aligned} p^{(1)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(1)} \cdot x_i x_j + \sum_{i=1}^{n} p_i^{(1)} \cdot x_i + p_0^{(1)} \\ p^{(2)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(2)} \cdot x_i x_j + \sum_{i=1}^{n} p_i^{(2)} \cdot x_i + p_0^{(2)} \\ &\vdots \\ p^{(m)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(m)} \cdot x_i x_j + \sum_{i=1}^{n} p_i^{(m)} \cdot x_i + p_0^{(m)} \end{aligned} \tag{1}$$

The security of multivariate schemes is based on the

MQ Problem: Given m multivariate quadratic polynomials $p^{(1)}(\mathbf{x}),\ldots,p^{(m)}(\mathbf{x})$ in n variables $x_1,\ldots,x_n$ as shown in equation (1), find a vector $\bar{\mathbf{x}} = (\bar{x}_1,\ldots,\bar{x}_n)$ such that $p^{(1)}(\bar{\mathbf{x}}) = \ldots = p^{(m)}(\bar{\mathbf{x}}) = 0$.

The MQ problem (for $m \approx n$) is proven to be NP-hard even for quadratic polynomials over the field GF(2) [12].

To build a public key cryptosystem on the basis of the MQ problem, one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ (central map). To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine (or linear) maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. The *public key* of the scheme is therefore given by $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$. The *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert the public key.

Note: Due to the above construction, the security of multivariate public key schemes is not only based on the MQ-Problem, but also on the EIP-Problem ("Extended Isomorphism of Polynomials") of finding the composition of $\mathcal{P}$.

In this paper we concentrate on multivariate signature schemes. The standard signature generation and verification process of a multivariate signature scheme works as shown in Figure 1.

Signature generation: To generate a signature for a message d, the signer uses a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ and computes recursively $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$. The signature of the message $\mathbf{w}$ is $\mathbf{z} \in \mathbb{F}^n$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of possibly many) pre-image of $\mathbf{x}$ under the central map $\mathcal{F}$.

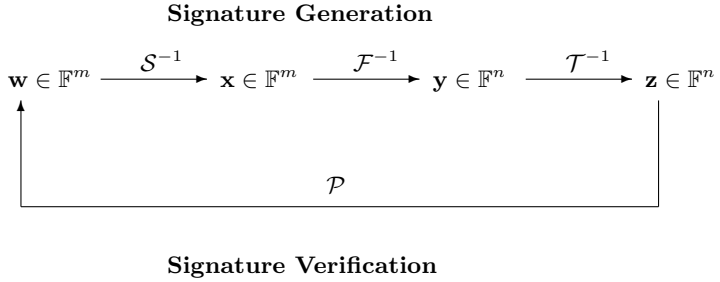

Fig. 1. General workflow of multivariate signature schemes

Verification: To check, if $\mathbf{z} \in \mathbb{F}^n$ is indeed a valid signature for a message d, one computes $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

A good overview of existing multivariate schemes can be found in [8].

3.1 The Rainbow Signature Scheme

The Rainbow signature scheme [9] is one of the most promising and best studied multivariate signature schemes. The scheme can be described as follows:
Let $\mathbb{F} = \mathbb{F}_q$ be a finite field with q elements, $n \in \mathbb{N}$ and $0 < v_1 < v_2 < \ldots < v_\ell < v_{\ell+1} = n$ be a sequence of integers. We set $m = n - v_1$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ and $V_i = \{1, \ldots, v_i\}$ $(i = 1, \ldots \ell)$.

Key Generation: The *private key* of the scheme consists of two invertible affine maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$ and a quadratic map $\mathcal{F}(\mathbf{x}) = (f^{(v_1+1)}(\mathbf{x}), \ldots, f^{(n)}(\mathbf{x})) : \mathbb{F}^n \to \mathbb{F}^m$. The polynomials $f^{(i)}$ $(i = v_1 + 1, \ldots, n)$ are of the form

$$f^{(i)} = \sum_{k,l \in V_j} \alpha_{k,l}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j, l \in O_j} \beta_{k,l}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j \cup O_j} \gamma_k^{(i)} \cdot x_k + \eta^{(i)} \quad (2)$$

with coefficients randomly chosen from $\mathbb{F}$. Here, j is the only integer such that $i \in O_j$. The *public key* is the composed map $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$.

Signature Generation To generate a signature for a document d, one uses a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ and computes recursively $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of approximately q^{v_1}) pre-image of $\mathbf{x}$ under the central map $\mathcal{F}$. This is done as shown in Algorithm 1.
It might happen that one of the linear systems in step 3 of the algorithm does

Algorithm 1 Inversion of the Rainbow central map

Input: Rainbow central map $\mathcal{F}$, vector $\mathbf{x} \in \mathbb{F}^m$
Output: vector $\mathbf{y} \in \mathbb{F}^n$ such that $\mathcal{F}(\mathbf{y}) = \mathbf{x}$

1: Choose random values for the variables $y_1, \ldots, y_{v_1}$ and substitute these values into the polynomials $f^{(i)}$ $(i = v_1 + 1, \ldots, n)$.
2: **for** $k = 1$ to ℓ **do**
3: Perform Gaussian Elimination on the polynomials $f^{(i)}$ $(i \in O_k)$ to get the values of the variables y_i $(i \in O_k)$.
4: Substitute the values of y_i $(i \in O_k)$ into the polynomials $f^{(i)}$ $(i \in \{v_{k+1} + 1, \ldots, n\})$.
5: **end for**

not have a solution. In this case one has to choose other values for $y_1, \ldots, y_{v_1}$ and start again.
The signature of the document d is $\mathbf{z} \in \mathbb{F}^n$.

Signature Verification To check, if $\mathbf{z} \in \mathbb{F}^n$ is indeed a valid signature for a document d, one computes $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

3.2 Multivariate Ring Signature Schemes

In the last years, a number of multivariate ring signature schemes have been proposed [19,31,28,27]. In this section, we give an overview of the main constructions and analyze them with regard to security and performance.

The schemes of Petzoldt et al. [19] and Zhang et al. [31]
These two schemes are threshold ring signature schemes, i.e. they allow the verifier to check if a minimal number s of users contributed to the signature ($1 \leq s \leq k$). Both of the schemes are based on the multivariate identification scheme of Sakumoto et al. [25], but use different techniques to extend the identification into a signature scheme: In the case of [19] this is the Fiat-Shamir protocol, the authors of [31] use the Γ-transformation. By both techniques it is possible to obtain a threshold ring signature scheme whose security is only based on the MQ Problem of solving a system of multivariate quadratic equations, which makes the schemes provable secure. However, due to the additional functionality of a *threshold* ring signature scheme, both schemes produce very long signatures. By restricting to a simple ring signature scheme (i.e. $s = 1$), we can reduce the signature length and improve the performance of the scheme drastically.

The scheme of L.L. Wang [27]
The ring signature scheme proposed by L.L. Wang in [27] is also based on the multivariate identification scheme of Sakumoto et al. [25]. Each user u_i $(u = 1, \ldots, k)$ chooses a vector $\mathbf{s}_i \in \mathbb{F}^n$ as his private key and a multivariate quadratic

system $\mathcal{P}_i : \mathbb{F}^n \to \mathbb{F}^m$ with $\mathcal{P}_i(\mathbf{s}_i) = 0$ as his public key. In order to generate a ring signature for a message d, the signer produces for each user a transcript of the identification scheme (using the "secret" $\mathbf{0}$ for the non signers). Unfortunately, the verifier has no means to check how many zero vectors were used during the signature generation. Therefore it is possible for an adversary which is no member of the ring and therefore does not know any of the private keys s_i to forge a valid ring signature (using $\mathbf{0}$ for all the secret vectors $\mathbf{s}_i$ $(i = 1, \ldots, k)$). Furthermore, the scheme proposed in [27] contains only one round of the identification scheme, enabling an adversary to forge a signature with probability $\frac{2}{3}$. Therefore, the scheme of [27] does not provide any security at all.

The scheme of S. Wang et al. [28]
The scheme of S. Wang et al. is similar to our construction in the sense that it provides a general technique to extend an arbitrary multivariate signature scheme to a ring signature scheme. Therefore, as it is in the case of our construction, the security of the resulting ring signature scheme is based on the security of the underlying multivariate signature scheme. However, in our construction, the ring signatures are generated in a much simpler and faster way. To generate a ring signature with the scheme of [28], one needs k hash function evaluations, $2k+1$ evaluations of public keys and one signature generation of the underlying signature scheme, while our scheme requires only $k-1$ evaluations of multivariate systems and one signature generation. During verification, [28] requires $k-1$ hash function evaluations and $2k-2$ evaluations of a multivariate quadratic system, while our scheme needs only k evaluations of the public key. Furthermore, these simple signature generation and verification algorithms make our scheme much easier to understand and to analyze and lead to (slightly) shorter ring signatures. Moreover, with regard to security, the paper [28] does not take attacks against underdetermined multivariate systems (see Section 4.1 and 5 of this paper) into consideration. Therefore, especially for large sizes of the group $\mathcal{R}$, the authors of [28] overestimate the security of their scheme significantly (or propose too small parameters).

4 Our Ring Signature Scheme

In this section we present our technique to extend arbitrary multivariate signature schemes such as UOV [14], Rainbow [9] and Gui [21] to ring signature schemes. Whereas, in this section, we present our technique in a very general way, we concentrate in the following sections on ring signatures based on the Rainbow signature scheme (see Section 3.1), which offers both good performance and short signatures. Furthermore, the key sizes of Rainbow are acceptable and can be further reduced by the technique of [18] (see Section 7).
Let $\mathcal{R} = \{u_1, \ldots, u_k\}$ be a ring of users.

Key Generation: Each user u_i generates a key pair $((\mathcal{S}_i, \mathcal{F}_i, \mathcal{T}_i), \mathcal{P}_i)$ of the underlying multivariate signature scheme. The public key $\mathcal{P}$ of the group is the

concatenation of all individual public keys, i.e. $\mathcal{P} = \mathcal{P}_1||\mathcal{P}_2||\ldots||\mathcal{P}_k$, while each user u_i keeps $\mathcal{S}_i, \mathcal{F}_i$ and $\mathcal{T}_i$ as his private key sk_i.

Signature Generation: In order to sign a message d on behalf of the ring $\mathcal{R}$, a 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. He then chooses random vectors $\mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_k \in \mathbb{F}^n$. He computes

$$\tilde{\mathbf{w}} = \mathbf{w} - \sum_{\substack{j=1 \\ j \neq i}}^{k} \mathcal{P}_j(\mathbf{z}_j) \in \mathbb{F}^m \tag{3}$$

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

Signature Verification: In order to check if $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \in \mathbb{F}^{k \cdot n}$ is indeed a valid ring signature for the message d, the receiver computes the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ of the message d and uses the public keys $\mathcal{P}_1, \ldots, \mathcal{P}_k$ to compute

$$\hat{\mathbf{w}} = \sum_{j=1}^{k} \mathcal{P}_j(\mathbf{z}_j). \tag{4}$$

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

Remark: In case of an honestly computed ring signature $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \in \mathbb{F}^{kn}$ we have

$$\hat{\mathbf{w}} = \sum_{j=1}^{k} \mathcal{P}_j(\mathbf{z}_j) = \sum_{\substack{j=1 \\ j \neq i}}^{k} \mathcal{P}_j(\mathbf{z}_j) + \mathcal{P}_i(\mathbf{z}_i) = \mathbf{w} - \tilde{\mathbf{w}} + \tilde{\mathbf{w}} = \mathbf{w}. \tag{5}$$

Therefore, an honestly generated ring signature is always accepted.

4.1 Security

In this section we analyze the security of our construction. We do not consider the security of the underlying multivariate signature schemes in this paper and refer to the original papers [14,21,17] for a security analysis of the different schemes. Here, we concentrate on our construction of a ring signature scheme. For this, we have to show the anonymity and unforgeability of the resulting scheme.

Anonymity

Theorem 1. *Our construction provides perfect anonymity for the actual signer as a member of the group, i.e. the final ring signature contains no information, which member of the group generated the signature and even a computationally unrestricted adversary can not reveal the identity of the signer.*

Proof (sketch): We assume that $\mathcal{R} = \{u_1, u_2\}$ and perform **Game[Anonymity]** (see Section 2) for this situation. Then we show that, independently of the fact which secret key is used during the generation of the ring signature, the signing oracle $\mathcal{OS}$ outputs each of the q^{n+v_1} possible ring signatures of the message $d^\star$ with probability $\approx q^{-n-v_1}$. For each possible ring signature $\sigma^\star$ of $d^\star$ we therefore have

$$\Pr[\sigma^\star \text{generated using } sk_1] = \Pr[\sigma^\star \text{ generated using } sk_2] \approx 1/2.$$

Therefore, an adversary can only guess whether $\sigma^\star$ was computed with sk_1 or sk_2 and his advantage is exactly 0 (independent from his resources).

Unforgeability To forge a ring signature with respect to a ring of signers $\mathcal{R} = \{u_1, \ldots, u_k\}$, an attacker has to find a solution $\mathbf{z}_1, \ldots, \mathbf{z}_k$ of the equation

$$\mathcal{P}_1(\mathbf{z}_1) + \mathcal{P}_2(\mathbf{z}_2) + \cdots + \mathcal{P}_k(\mathbf{z}_k) = \mathbf{w}. \tag{6}$$

Basically, there are two possibilities to do this.

1. The adversary could proceed similar to a legitimate user of the ring signature scheme and choose $k-1$ random vectors $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1} \in \mathbb{F}^n$, compute $\tilde{\mathbf{w}} = \mathbf{w} - \sum_{i=1}^{k-1} \mathcal{P}_i(\mathbf{z}_i)$ and try to find a solution to the system $\mathcal{P}_k(\mathbf{z}_k) = \tilde{\mathbf{w}}$.
2. The adversary could try to solve the system (6) directly as an underdetermined system of multivariate quadratic equations.

Note that the first case is equivalent to breaking an instance of the underlying multivariate signature scheme. We do not consider this case here and refer to the papers [14,21,17] for a security analysis of the various schemes, We assume that, if we choose the parameters of our scheme according to the recommendations given in these papers, our scheme is secure against attacks of this kind. Hence, we concentrate in the following on the second case.
Unfortunately, solving equation (6) directly is not as hard as breaking the underlying scheme, where the attacker has to find a solution $\mathbf{z_k} \in \mathbb{F}^n$ of $\mathcal{P}_\parallel(\mathbf{z_k}) = \tilde{\mathbf{w}}$. The reason for this is that the system (6) is a highly underdetermined multivariate quadratic system. For systems of this type we have to consider the following two important results.

1. If the number of variables n in an underdetermined 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 [26].
2. If the number of variables n in the multivariate quadratic system $\mathcal{P}$ exceeds $n \geq \frac{m(m+3)}{2}$, $\mathcal{P}$ can be solved in polynomial time [15].

In our parameter choice (see next section), we have to consider these two results. Therefore, the parameters of our scheme depend not only on the required level of security, but, since the number of variables in the public system $\mathcal{P}$ (6) depends on k, also on the size of the ring $\mathcal{R}$.

5 Parameters

In this section we give concrete parameter proposals for our ring signature scheme. We define our scheme over the field GF(256) and instantiate it on the basis of the Rainbow signature scheme of Section 3.1, which offers both good performance and short signatures. The proposed parameter sets are obtained as follows.

1. Direct attacks against the scheme should be infeasible, i.e. the parameters of the scheme have to be chosen in such a way that the two attacks against underdetermined quadratic systems mentioned in the previous section become infeasible.
2. Attacks of the Rainbow type against the single systems $\mathcal{P}(\mathbf{z}_i) = \mathbf{w}_i$ $(i = 1, \ldots k)$ must be infeasible. With regard to this, we follow the results of [17].

As we find, for small numbers of k (e.g. $k = 5$), the parameters of our scheme are very similar to the parameters recommended for Rainbow in [17]. For larger values of k, attacks against underdetermined systems play an increasing role. The resulting parameter sets and key sizes can be found in Table 1.

security level (bit)		5 users	10 users	20 users	50 users
80	parameters	(16,17,15)	(15,20,18)	(14,26,24)	(13,56,53)
	public key size (kB)	191	551	2,095	40,588
	signature size (bit)	1,920	4,240	10,240	48,800
100	parameters	(25,21,19)	(24,25,22)	(22,31,28)	(20,60,55)
	public key size (kB)	432	1,206	3,921	52,312
	signature size (bit)	2,600	5,680	12,960	54,000
128	parameters	(36,23,20)	(34,26,23)	(32,33,29)	(30,64,58)
	public key size (kB)	680	1,708	5,522	70,180
	signature size (bit)	3,160	6,640	15,040	60,800

Table 1. Proposed Parameters for our Ring Signature Scheme ($\mathbb{F} = \mathrm{GF}(256)$; Rainbow)

As the table shows, especially for small values of k, the signature sizes of our scheme are quite small. The size of a ring signature is of range several kbit and therefore not longer than standard signatures of many other post-quantum (e.g. lattice, hash based) signature schemes. However, for larger values of k, key and signature sizes of our scheme increase significantly.

6 Alternative Construction of a Multivariate Ring Signature Scheme

As can be seen from Table 1, the key sizes (especially the size of the public key) increase drastically if the number of users in the ring gets larger. To avoid this,

we present in this section an alternative way to construct a ring signature scheme on the basis of multivariate signature schemes such as Rainbow. In particular, we use here instead of component wise addition of the single signatures component wise multiplication. By doing so, we can prevent attacks against highly underdetermined multivariate quadratic systems, since the degree of the corresponding system becomes very large. Our alternative construction can be described as follows.

Key Generation: The key generation of our alternative construction works just as presented in Section 4. Each user u_i generates a key pair $((\mathcal{S}_i, \mathcal{F}_i, \mathcal{T}_i), \mathcal{P}_i)$ of the underlying multivariate signature scheme. The public key $\mathcal{P}$ of the group is the set of all individual public keys, i.e. $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k\}$, while each user u_i keeps $\mathcal{S}_i, \mathcal{F}_i$ and $\mathcal{T}_i$ as his private key sk_i.

Signature Generation: In order to sign a message d on behalf of the ring $\mathcal{R}$, a user u_i uses a hash function $\mathcal{H} : \{0,1\}^\star \to \{0, \ldots, q-1\}^m$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) + 1^m \in \mathbb{F}^m$ of the message, where 1^m is a vector with all entries equal to one. He then chooses random vectors $\mathbf{z}_1, \ldots, \mathbf{z}_{i-1}, \mathbf{z}_{i+1}, \ldots, \mathbf{z}_k \in \mathbb{F}^n$ satisfying

$$(\mathcal{P}_j(\mathbf{z}_j))_s \neq 0, \quad j \in \{1, \ldots, k\} \setminus \{i\}, \ s \in \{1, \ldots, m\}.$$

He computes

$$\tilde{\mathbf{w}} = \mathbf{w} \cdot (\prod_{\substack{j=1 \\ j \neq i}}^{k} \mathcal{P}_j(\mathbf{z}_j))^{-1} \in \mathbb{F}^m \tag{7}$$

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 for the message d is $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \in \mathbb{F}^{kn}$. Note that in equation (7) multiplication and inversion work component wise on the elements of the corresponding vectors.

Remark: The reason of constructing the hash value of the document d in the way shown above is to generate a hash value without zero elements. By doing so we can ensure that all vectors $\mathbf{w}_i$ have the same structure. This guarantees the anonymity of the actual signer.

Signature Verification: In order to check if $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \in \mathbb{F}^{kn}$ is indeed a valid ring signature for the message d, the receiver computes the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ of the message d and uses the public keys $\mathcal{P}_1, \ldots, \mathcal{P}_k$ to compute

$$\hat{\mathbf{w}} = \prod_{j=1}^{k} \mathcal{P}_j(\mathbf{z}_j). \tag{8}$$

If $\hat{\mathbf{w}} = \mathbf{w}$ holds, the signature is accepted, otherwise it is rejected. Again note that the multiplication works component wise.

Remark: In the case of an honestly computed ring signature $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) \in \mathbb{F}^{kn}$ we have

$$\hat{\mathbf{w}} = \prod_{j=1}^{k} \mathcal{P}_j(\mathbf{z}_j) = \prod_{\substack{j=1 \\ j \neq i}}^{k} \mathcal{P}_j(\mathbf{z}_j) \cdot \mathcal{P}_i(\mathbf{z}_i) = \mathbf{w} \cdot (\tilde{\mathbf{w}})^{-1} \cdot \tilde{\mathbf{w}} = \mathbf{w}. \tag{9}$$

Therefore, an honestly generated ring signature is always accepted.

6.1 Unforgeability

While the anonymity of our ring signature scheme can be shown exactly as in Section 4.1., we here concentrate on the unforgeability. Similar to Section 4.1, an attacker can try to forge a ring signature in two different ways:

1. The adversary could proceed similar to a legitimate user of the ring signature scheme and choose $k-1$ random vectors $\mathbf{z}_1, \ldots, \mathbf{z}_{k-1} \in \mathbb{F}^n$, compute $\tilde{\mathbf{w}} = \mathbf{w} \cdot (\prod_{j=1}^{k-1} \mathcal{P}_j(\mathbf{z}_j))^{-1}$ and try to find a solution of the system $\mathcal{P}_k(\mathbf{z}_k) = \tilde{\mathbf{w}}$.
2. The adversary could try to solve the system

   $$\mathcal{P}_1(\mathbf{z}_1) \cdot \ldots \cdot \mathcal{P}_k(\mathbf{z}_k) = \mathbf{w}$$

   directly as an underdetermined system of multivariate equations.

Again, forging a ring signature by the first method is equivalent to breaking an instance of the underlying multivariate scheme, which is, by our assumption, infeasible.
When attacking our scheme in the second way, the attacker is faced with an underdetermined system of multivariate polynomial equations. But, in contrast to Section 4.1, this system is no longer quadratic, but the polynomials are, for a ring of k users, of degree $2k$. The methods to solve underdetermined quadratic systems mentioned in Section 4.1 do not work in this case[3]. It is therefore infeasible for the attacker to forge a ring signature using this strategy. This means that we do not have to increase the parameters of our scheme when the number of users in the ring gets large. Beyond the significant reduction of key size this also makes it much easier to add additional users to the ring.
Table 2 shows our parameter recommendations and resulting key and signature sizes for our alternative construction of a multivariate ring signature scheme.

[3] Of course, the attacker could try to transform the given system of high degree into a quadratic one. However, even if the given system is very sparse, this increases the number of equations and variables in the quadratic system drastically. Furthermore, the ratio between the number of variables and the number of equations gets close to 1.

security level (bit)		5 users	10 users	20 users	50 users
80	parameters (v_1, o_1, o_2)	(17,13,13)	(17,13,13)	(17,13,13)	(17,13,13)
	public key size (kB)	125.7	251.4	502.7	1,257
	signature size (bit)	1,720	3,440	6,880	17,200
100	parameters (v_1, o_1, o_2)	(26,16,17)	(26,16,17)	(26,16,17)	(26,16,17)
	public key size (kB)	294.9	589.7	1,179	2,949
	signature size (bit)	2,6360	4,720	9,440	23,600
128	parameters (v_1, o_1, o_2)	(36,21,22)	(36,21,22)	(36,21,22)	(36,21,22)
	public key size (kB)	680.3	1,361	2,721	6,803
	signature size (bit)	3,160	6,320	12,640	31,600

Table 2. Proposed Parameters for our alternative Construction of a multivariate Ring Signature Scheme ($\mathbb{F}$ = GF(256); Rainbow)

7 Reduction of Public Key Size

In [18], Petzoldt et al. proposed a technique to reduce the public key size of the UOV and Rainbow signature schemes. In particular, they were able to construct a Rainbow key pair $((\mathcal{S}, \mathcal{F}, \mathcal{T}), \mathcal{P})$, where the coefficient matrix P of the public key has the form (for Rainbow schemes with two layers)

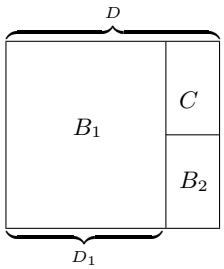

Fig. 2. Structure of the public key $\mathcal{P}$

Here we have $D = \frac{(n+1)\cdot(n+2)}{2}$ and $D_1 = D - \frac{(o_2+1)\cdot(o_2+2)-2}{2}$. The matrices $B_1 \in \mathbb{F}^{m \times D_1}$ and $B_2 \in \mathbb{F}^{o_2 \times (D-D_1)}$ can be arbitrarily set by the user. In particular, B_1 and B_2 can be chosen in a structured way which reduces the public key size of the Rainbow scheme significantly.
Note that, when applying this technique to our ring signature scheme, we can choose the same matrices B_1 and B_2 for all users $u_1, \ldots, u_k$. By doing so, we can reduce the public key size of our scheme by up to 68 % (see Table 3).

Furthermore, when choosing the matrices B_1 and B_2 in a cyclic way, we can speed up the evaluation of the Rainbow public key by up to 60 % [20]. Since this step is used both in the signature generation and verification processes of our scheme, both processes can be sped up drastically (see Table 4).

security level (bit)		5 users	10 users	20 users	50 users
80	parameters	(17,13,13)	(17,13,13)	(17,13,13)	(17,13,13)
	public key size (standard) (kB)	125.7	251.4	502.7	1,257
	public key size (reduced) (kB)	47.3	93.8	186.7	465.5
	reduction (%)	62.4	62.7	62.9	63.0
100	parameters	(26,16,17)	(26,16,17)	(26,16,17)	(26,16,17)
	public key size (standard) (kB)	294.9	589.7	1,179	2,949
	public key size (reduced) (kB)	99.5	197.3	393.0	980.2
	reduction (%)	66.3	66.5	66.7	66.8
128	parameters	(36,21,22)	(36,21,22)	(36,21,22)	(36,21,22)
	public key size (standard) (kB)	680.3	1,361	2,721	6,803
	public key size (reduced) (kB)	219.4	435.9	868.8	2,168
	reduction (%)	67.7	68.0	68.1	68.1

Table 3. Possible Reduction of Public Key Size

8 Implementation and Efficiency Results

In this section we present our results regarding the efficiency of our construction. To generate a ring signature on behalf of a ring $\mathcal{R} = \{u_1, \ldots, u_k\}$ of k members, a user u_i has to perform

- $k - 1$ evaluations of public systems $\mathcal{P}_i$ and
- 1 Rainbow signature generation.

The verification process of our scheme consists of k evaluations of the Rainbow public keys $\mathcal{P}_1, \ldots, \mathcal{P}_k$.
Since both the evaluation and inversion of Rainbow systems are very efficient, our scheme offers good performance. By using structured public keys (see last section), we can speed up our scheme further (see Table 4).
To study the efficiency of our construction in practice, we created a straightforward C implementation of Rainbow and our ring signature scheme and ran it for the parameter sets proposed in Section 5. Table 4 shows the results. In each cell, the first number shows the running time of the signature generation / verification process of the standard scheme, while the second number shows the corresponding timings for the structured scheme.

9 Discussion

Especially for small values of k, our ring signature scheme is very efficient. In this case, the resulting ring signatures are not larger than standard signatures of other post-quantum signature schemes such as lattice and hash based constructions.
However, when the number of users in the ring $\mathcal{R}$ increases, key sizes and signature sizes of our scheme increase significantly.

security level (bit)		5 users	10 users	20 users	50 users
80	parameters	(16,17,15)	(15,20,18)	(14,26,24)	(13,56,53)
	sign. generation (ms)	13.31 / 9.15	28.73 / 17.31	58.98 / 37.04	1225 / 738.0
	sign. verification (ms)	10.81 / 5.08	26.23 / 13.54	50.51 / 27.42	1200 / 703.0
100	parameters	(25,21,19)	(24,25,22)	(22,31,28)	(20,60,55)
	sign. generation (ms)	16.04 / 10.30	41.07 / 23.81	123.80 / 68.10	1580.32 / 905.4
	sign. verification (ms)	12.75 / 5.57	35.55 / 16.38	115.97 / 57.35	1547.69 / 859.0
128	parameters	(36,23,20)	(34,26,23)	(32,33,29)	(30,64,58)
	sign. generation (ms)	27.56 / 17.54	58.97 / 32.00	175.37 / 88.94	3177.13 / 1706
	sign. verification (ms)	20.03 / 7.70	50.43 / 20.45	163.22 / 72.24	3110.79 / 1610

Table 4. Running Times of Signature Generation and Verification

Another disadvantage of the first version of our scheme is that it is very difficult to add additional users to the ring $\mathcal{R}$, since this might made it necessary to change the parameters. One therefore has to fix the maximal number $k_{\max}$ of users in the ring $\mathcal{R}$ a priori and choose the parameters of the scheme according to $k_{\max}$. This problem can be solved by switching from addition to componentwise multiplication (see Section 6). Table 5 shows a comparison of our scheme with other (post-quantum and classical) ring signature schemes.

	scheme	our	[29]	[16]	[7]	[1]	[2]	[10]
5 users	pk size (kB)	47.3	1.3	179	147	751	1.0	1.0
	sign. size (bit)	1,720	26,000	301,546	659,632	6,973,251	7,810	15,820
50 users	pk size (kB)	465.5	12.5	1,785	7,513	15,020	9.8	9.8
	sign. size (bit)	17,200	260,000	3,015,462	6,596,320	69,732,510	78,100	158,200
		mult.	lattice	mult.	lattice	code	RSA	DL
				Threshold Ring Signatures				

Table 5. Comparison of our scheme with other ring signature schemes (80 bit security)

As the Table shows, our scheme outperforms the other post-quantum ring signature schemes in terms of signature size. In this sense, it can also compete with the RSA and DL based constructions of [2] and [10]. On the other hand, the key sizes of our scheme are much larger than those of the classical and the lattice based construction of [29].
Furthermore, there exist some pairing based constructions of ring signature schemes, which offer a sublinear signature size [11]. For large values of k, the ring signatures of these schemes are significantly smaller than those of the above constructions. However, these schemes can be easily broken by quantum computers. Therefore, our scheme offers the shortest ring signatures of all post-quantum constructions.

10 Conclusion

In this paper we proposed a new multivariate ring signature scheme on the basis of the Rainbow signature scheme [9]. However, we can construct our scheme on the basis of every other multivariate signature scheme such as UOV and HVEv-, too. Our scheme is one of the first multivariate signature schemes with special properties and one of few candidates for post-quantum ring signatures. The scheme is very efficient, especially when the number of users in the ring is small, and produces the shortest ring signatures of all existing post-quantum constructions.
Future work includes the development of other multivariate signature schemes with special properties such as blind and group signatures.

Acknowledgements

The authors would like to thank the reviewers in particular for their helpful comments. The first author is supported by the European Commission through the Horizon 2020research and innovation programme under grant agreement No H2020-ICT-2014-645622 PQCRYPTO.

References

1. C. Aguilar, P.L. Cayrel, P. Gaborit and F. Laguillaumie: A New Efficient Threshold Ring Signature Scheme Based on Coding Theory. IEEE Transactions on Information Theory 57(7), pp. 4833-4842 (2011).
2. M. R. Asaar, M. Salmasizadeh, W. Susilo: A short identity-based proxy ring signature scheme from RSA. Computer Standards and Interfaces 38:144–151 (2015).
3. A. Bender, J. Katz, R. Morselli: Ring Signatures: Stronger Definitions and Constructions without random Oracles. IACR eprint 2005/304.
4. D.J. Bernstein, J. Buchmann, E. Dahmen (eds.): Post Quantum Cryptography. Springer, 2009.
5. A. Bogdanov, T. Eisenbarth, A. Rupp, C. Wolf: Time-area optimized public-key engines: MQ-cryptosystems as replacement for elliptic curves? CHES 2008, LNCS vol. 5154, pp. 45-61. Springer, 2008.
6. A.I.T. Chen, M.-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E. L.-H. Kuo, F. Y.-S. Lee, B.-Y. Yang: SSE implementation of multivariate PKCs on modern x86 cpus. CHES 2009, LNCS vol. 5747, pp. 33 - 48. Springer, 2009.
7. P.L. Cayrel, R. Lindner, M. Rückert and R. Silva: A Lattice-Based Threshold Ring Signature Scheme. LATINCRYPT 2010, LNCS vol. 6212, pp. 255 - 272, Springer, 2010.
8. J. Ding, J. E. Gower, D. S. Schmidt: Multivariate Public Key Cryptosystems. Springer, 2006.
9. J. Ding, D. S. Schmidt: Rainbow, a new multivariate polynomial signature scheme. ACNS 2005, LNCS vol. 3531, pp. 164-175. Springer, 2005.
10. M. Franklin, H. Zhang: Unique Ring Signatures: A Practical Construction. Financial Cryptography, LNCS vol. 7859, pp. 162 - 170. Springer, 2013.

11. E. Fujisaki: Sub-linear size traceable ring signatures without random oracles. CTRSA, LNCS vol. 6558, pp. 393 - 415. Springer 2011.
12. M. R. Garey and D. S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company 1979.
13. D. Kravitz: Digital Signature Algorithm. US patent 5231668 (July 1991).
14. A. Kipnis, L. Patarin, L. Goubin: Unbalanced Oil and Vinegar Schemes. EUROCRYPT 1999, LNCS vol. 1592, pp. 206–222. Springer, 1999.
15. H. Miura, Y. Hashimoto, T. Takagi: Extended Algorithm for Solving Underdefined Multivariate Quadratic Equations. PQCrypto 2013. LNCS vol. 7932, pp. 118 - 135. Springer, 2013.
16. A. Petzoldt, S. Bulygin, J. Buchmann: A Multivariate based Threshold Ring Signature Scheme. Appl. Algebra Eng. Commun. Comput. 24(3-4); 255-275 (2012).
17. A. Petzoldt, S. Bulygin, J. Buchmann: Selecting Parameters for the Rainbow Signature Scheme. PQCrypto 2010, LNCS vol. 6061, pp. 218-240. Springer, 2010.
18. A. Petzoldt, S. Bulygin, J. Buchmann: CyclicRainbow - A Multivariate Signature Scheme with a Partially Cyclic Public Key. INDOCRYPT 2010, LNCS vol. 6498, pp. 33-48. Springer, 2010.
19. A. Petzoldt, S. Bulygin, J. Buchmann: A Multivariate Threshold Ring Signature Scheme. AAECC 25 (3-4), pp. 255 - 275 (2012).
20. A. Petzoldt, S. Bulygin, J. Buchmann: Fast Verification for Improved Versions of the UOV and Rainbow Signature Schemes. PQCrypto, LNCS vol. 7932, pp. 188-202. Springer, 2013.
21. A. Petzoldt, M.S. Chen, B.Y. Yang, C. Tao, J. Ding: Design Principles for HFEv-based Signature Schemes. ASIACRYPT 2015 - Part 1, LNCS vol. 9452, pp. 311-334. Springer, 2015.
22. R. L. Rivest, A. Shamir, L. Adleman: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 21 (2), pp. 120-126 (1978).
23. R. L. Rivest, A. Shamir, Y. Taumann: How to leak a secret. ASIACRYPT 2001, LNCS vol. 2248, pages 552 - 565. Springer, 2001.
24. P. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. Comput. 26 (5), pp. 1484 - 1509 (1997).
25. K. Sakumoto, T. Shirai and H. Hiwatari: Public-Key Identification Schemes based on Multivariate Quadratic Polynomials. CRYPTO 2011, LNCS vol. 6841, pp. 706 - 723, Springer 2011.
26. E. Thomae, C. Wolf: Solving Underdetermined Systems of Multivariate Quadratic Equations Revisited. PQCrypto 2012, LNCS vol. 7293, pp. 156–171. Springer, 2012.
27. L.L. Wang: A New Multivariate-based Ring Signature Scheme. Proceeedings of ISCCCA 2013.
28. S. Wang, R. Ma, Y. Zhang, X. Wang: Ring signature scheme based on multivariate public key cryptosystems. Computers and Mathematics with Applications 62 (2011) 3973-3979.
29. S. Wang, R. Zhao: Lattice-Based Ring Signature Scheme under the Random Oracle Model, CoRR, vol. abs/1405.3177 (2014)
30. B.Y. Yang, J.M. Chen: Building secure tame-like multivariate public-key cryptosystems.: The new TTS. CHES 2004, LNCS vol. 3156, pp. 371- 385. Springer, 2004.
31. J. Zhang, Y. Zhao: A New Multivariate Based Threshold Ring Signature Scheme. NSS 14, LNCS vol. 8792, pp. 526 - 533. Springer 2014.