The Shortest Signatures Ever

Mohamed Saied Emam Mohamed[1], Albrecht Petzoldt[2]

[1] Technische Universität Darmstadt, Germany
[2] Kyushu University, Fukuoka, Japan

mohamed@cdc.informatik.tu-darmstadt.de, petzoldt@imi.kyushu-u.ac.jp

Abstract. Multivariate Cryptography is one of the main candidates for creating post quantum public key cryptosystems. Especially in the area of digital signatures, there exist many practical and secure multivariate schemes. In this paper we present a general technique to reduce the signature size of multivariate schemes. Our technique enables us to reduce the signature size of nearly all multivariate signature schemes by 10 to 15 % without slowing down the scheme significantly. We can prove that the security of the underlying scheme is not weakened by this modification. Furthermore, the technique enables a further reduction of the signature size when accepting a slightly more costly verification process. This trade off between signature size and complexity of the verification process can not be observed for any other class of digital signature schemes. By applying our technique to the Gui signature scheme, we obtain the shortest signatures of all existing digital signature schemes.

Keywords: Post Quantum Cryptography, Multivariate Cryptography, Digital Signatures, Signature Size

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 [18], DSA [10] and ECC. However, schemes like these will become insecure as soon as large enough quantum computers arrive. The reason for this is Shor's algorithm [19], 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). The increasing importance of research in this area has recently been emphasized by a number of authorities. For example, the american National Security Agency (NSA) has recommended governmental organizations to change their security infrastructures from schemes like RSA to post quantum schemes [13] and the National Institute of Standards and Technologies (NIST) is preparing to standardize these schemes [14].

According to [14], one of the main candidates for this standardization is multivariate cryptography. 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 [2,3]. Additionally, at least in the area of digital signatures, there exists a large number of practical multivariate schemes [6,11,17].

In this paper we present a general technique to reduce the signature size of multivariate signature schemes. The key idea of our strategy is to send only a part of the signature to the receiver of a signed message. The verification process consists in solving a multivariate quadratic system in a very small number of variables.

By our technique we can reduce the signature size of nearly every multivariate signature scheme by 10 to 15 % without increasing the key sizes or slowing down the scheme significantly. Furthermore, we can prove that the security of the signature schemes is not weakened by our modifications. Moreover, we can achieve a further reduction of the signature length when accepting a small increase of the verification cost. This trade off is, to our knowledge, unique for multivariate schemes and can not be observed for any other class of digital signature schemes. By applying our technique to the Gui signature scheme of Asiacrypt 2015 [17], we can reduce the signature size of this scheme to 110 bit (80 bit security), by which we obtain the shortest signatures of all existing schemes.

Our technique is especially attractive in situations, in which the connection between the sender and the receiver is very slow. An example for this are (wireless) sensor networks. In such systems, the actual messages are often very short and therefore a major part of the communication consists of the signatures itself. Reducing the signature length therefore reduces the communication cost and speeds up the system significantly.

The rest of this paper is organized as follows. In Section 2, we give a short overview of (wireless) sensor networks and show how these systems can be sped up by our technique. Section 3 gives an overview of the basic concepts of multivariate cryptography and introduces two of the best known and most widely studied multivariate signature schemes, namely the Rainbow and the HFEv- signature schemes. Section 4 describes the most important methods for solving systems of multivariate quadratic equations which can be used in the verification process of our schemes. In Section 5 we present our technique in detail and show that our modifications do not weaken the security of the underlying schemes. Section 6 shows, for concrete parameter sets of Rainbow and HFEv-, which actual reductions in terms of the signature size can be achieved by our technique. Furthermore, in this section, we analyze the efficiency of our technique using a large set of experiments. In Section 7 we show how to apply our technique to the Gui signature scheme, which allows us to efficiently generate secure signatures of size only 110 bit (80 bit security). Finally, Section 8 gives a short discussion of the benefits of our technique and Section 9 concludes the paper.

2 (Wireless) Sensor Networks

In this section we describe a possible application scenario for our technique. Since we aim at reducing the signature size, our technique is especially attractive in situations in which the connection between sender and receiver is very slow. An example for this are (wireless) sensor networks.

Let us assume that we have a number of small sensors, which report at regular intervals their status to a server (as shown in Figure 1). The status messages itself might be very short, but the messages have to be signed in order to prevent adversaries to send false

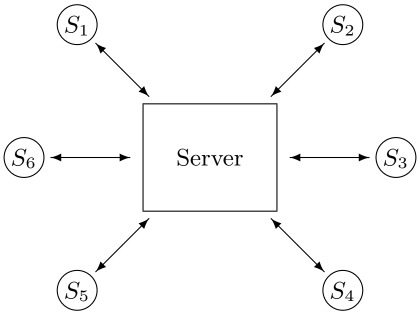

Fig. 1. Sensor Network

messages to the server. The single sensors have only restricted computing, power and memory capacities. Therefore, multivariate signature schemes are an attractive candidate to generate the above mentioned signatures. To do this, one has to store the private key of the multivariate scheme on the sensor. For current multivariate schemes such as Gui, the size of the private key is only 3kB. If this is still too big for the sensor, the private key can easily be stored as a random seed.
However, the main problem in our scenario is the slow connection between the sensors and their server. Therefore, our goal is to reduce the communication between the sensors and the server as far as possible. Since, in the upper status messages, the message itself might be very short, the length of the signature plays a major role. Now, multivariate scheme already generate relatively short signatures (usually a few hundred bits). However, as we will show in Section 5, we can reduce the length of multivariate signatures further, without weakening the security of the scheme, increasing the key sizes or making the signature generation process more costly. By doing so, we can reduce the amount of communication and therefore speed up the system significantly.

3 Multivariate Cryptography

The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials. 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 $x_1, \ldots, x_n$, 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) [8].

To build a public key cryptosystem based on the MQ problem, one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \rightarrow \mathbb{F}^m$ (central map). To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine (or linear) maps $\mathcal{S} : \mathbb{F}^m \rightarrow \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \rightarrow \mathbb{F}^n$. The *public key* of the scheme is therefore given by $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \rightarrow \mathbb{F}^m$. 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 cryptosystems is not only based on the MQ-Problem but also on the EIP-Problem ("Extended Isomorphism of Polynomials") of finding the composition of $\mathcal{P}$ [5].

In this paper we concentrate on multivariate signature schemes. For this we require $n \geq m$, which ensures that every message has a signature. The standard signature generation and verification process of a multivariate signature scheme works as shown in Figure 2.

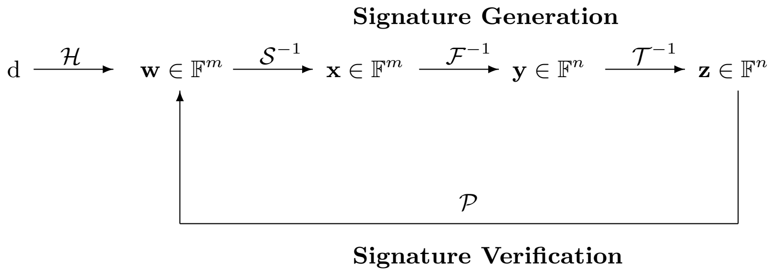

Fig. 2. General workflow of multivariate signature schemes

Signature generation: To generate a signature for a document d, we use a hash function $\mathcal{H}$ to compute a hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$. After that, one 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 document d is $\mathbf{z} \in \mathbb{F}^n$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of approximately q^{n-m}) pre-image of $\mathbf{x}$ under the central map $\mathcal{F}$.

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

Since the late 1980's, many multivariate schemes both for encryption and digital signatures have been proposed. The first one was the Matsumoto-Imai cryptosystem [12], which was later extended to schemes like Sflash [16] and HFE [15]. A different research direction lead to the development of multivariate schemes such as UOV [11], Rainbow [6], enTTS [21] and SimpleMatrix [20]. Although several of these schemes have been broken due to newly developed attacks, a number of multivariate schemes such as UOV, Rainbow and HFEv- has withstood (for suitable parameter sets) cryptanalysis for more than 20 years now. In the next two subsections, we introduce two of these schemes.

3.1 The Rainbow Signature Scheme

The Rainbow signature scheme [6] is one of the most promising and best studied multivariate schemes. The scheme can be described as follows.
Let $\mathbb{F}$ be a finite field, $n \in \mathbb{N}$ and $v_1 < v_2 < \ldots < v_\ell < v_{\ell+1} = n$ be a sequence of integers. We set $O_i = \{v_i + 1, \ldots, v_{i+1}\}$ 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 $\mathcal{S} : \mathbb{F}^m \rightarrow \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \rightarrow \mathbb{F}^n$ and a quadratic map $\mathcal{F}(\mathbf{x}) = (f^{(v_1+1)}(\mathbf{x}), \ldots, f^{(n)}(\mathbf{x})) : \mathbb{F}^n \rightarrow \mathbb{F}^m$. Here, $m = n - v_1$ is the number of components of $\mathcal{F}$.
The components of the central map $\mathcal{F}$ are of the form

$$f^{(i)} = \sum_{k,l \in V_j} \alpha_{kl}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j, l \in O_j} \beta_{kl}^{(i)} \cdot x_k \cdot x_l + \sum_{k \in V_j \cup O_j} \gamma_k^{(i)} \cdot x_k + \eta^{(i)}. \qquad (1)$$

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

Signature Generation: To generate a signature for a document d, one uses a hash function $\mathcal{H}$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$. After that, one computes recursively $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w})$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x})$ 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 $\mathcal{F}$. In the case of Rainbow, this is done as follows.

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 $\ell$ 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
```

It might happen that one of the linear systems in step 3 of the algorithm does 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 the document d, one computes $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

3.2 The HFEv- Signature Scheme

Another widely known construction is the HFEv- signature scheme, which is often used as the basis of more advanced multivariate signature schemes such as QUARTZ and Gui [17] (see Section 7).
Let $\mathbb{F} = \mathbb{F}_q$ be a finite field with q elements and $\mathbb{E}$ be a degree n extension field of $\mathbb{F}$. Furthermore, we choose integers D, a and v. Let Φ be the canonical isomorphism between $\mathbb{F}^n$ and $\mathbb{E}$, i.e.

$$\Phi(x_1, \ldots, x_n) = \sum_{i=1}^{n} x_i \cdot X^{i-1}. \tag{2}$$

The central map $\mathcal{F}$ of the HFEv- scheme is a map from $\mathbb{F}^v \times \mathbb{E}$ to $\mathbb{E}$ of the form

$$\mathcal{F}(X) = \sum_{0 \le i \le j}^{q^i + q^j \le D} \alpha_{ij} \cdot X^{q^i + q^j} + \sum_{i=0}^{q^i \le D} \beta_i(v_1, \ldots, v_v) \cdot X^{q^i} + \gamma(v_1, \ldots, v_v), \tag{3}$$

with $\alpha_{ij} \in \mathbb{E}$, $\beta_i : \mathbb{F}^v \to \mathbb{E}$ being linear and $\gamma : \mathbb{F}^v \to \mathbb{E}$ being a quadratic function.
Due to the special form of $\mathcal{F}$, the map $\bar{\mathcal{F}} = \Phi^{-1} \circ \mathcal{F} \circ \Phi$ is a multivariate quadratic map from $\mathbb{F}^{n+v}$ to $\mathbb{F}^n$. To hide the structure of $\bar{\mathcal{F}}$ in the public key, one combines it with two affine (or linear) maps $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^{n-a}$ and $\mathcal{T} : \mathbb{F}^{n+v} \to \mathbb{F}^{n+v}$ of maximal rank.

The *public key* of the scheme is the composed map $\mathcal{P} = \mathcal{S} \circ \bar{\mathcal{F}} \circ \mathcal{T} : \mathbb{F}^{n+v} \to \mathbb{F}^{n-a}$, the *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$.

Signature generation: To generate a signature for a document d, we use a hash function $\mathcal{H}$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^{n-a}$. After that, the signer performs the following three steps.

1. Compute a preimage $\mathbf{x} \in \mathbb{F}^n$ of $\mathbf{w}$ under the affine map $\mathcal{S}$.
2. Lift $\mathbf{x}$ to the extension field $\mathbb{E}$ (using the isomorphism Φ). Denote the result by X.
   Choose random values for the Vinegar variables $v_1, \ldots, v_v \in \mathbb{F}$ and compute $\mathcal{F}_V = \mathcal{F}(v_1, \ldots, v_v)$.
   Solve the univariate polynomial equation $\mathcal{F}_V(Y) = X$ by Berlekamp's algorithm and compute $\mathbf{y}' = \Phi^{-1}(Y) \in \mathbb{F}^n$. Set $\mathbf{y} = (\mathbf{y}'||v_1|| \ldots ||v_v)$.
3. Compute the signature $\mathbf{z} \in \mathbb{F}^{n+v}$ of the document d by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Signature verification: To check if $\mathbf{z} \in \mathbb{F}^{n+v}$ is indeed a valid signature for the document d, we compute the hash value $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^{n-a}$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

4 Solving multivariate quadratic systems

In this section we give a short overview of the most important techniques for solving systems of multivariate quadratic equations that might be used in our scheme.

4.1 The Relinearization Technique

In [11], Kipnis and Patarin proposed the Relinearization technique, which allows to solve highly overdetermined multivariate quadratic systems in polynomial time. In particular, a system can be solved by this technique if the number of equations m is greater or equal to

$$m \geq \frac{(n+1) \cdot (n+2)}{2} - 1. \tag{4}$$

The idea can be described as follows:

1. Interpret each of the quadratic monomials $x_i \cdot x_j$ in the system as a new variable x_{ij}.
2. Solve the resulting linear system by Gaussian Elimination.

If equation (4) holds, the linear system in step 2 has (in most cases) exactly one solution, which directly yields the solution of the quadratic system.

4.2 Other techniques

If the number n of variables in the system exceeds the upper bound given by equation (4), the Relinearization method produces a huge set of fake solutions, which are solutions to the linear system, but do not solve the original quadratic one. In this case one has to use other methods to solve the quadratic system, for example XL (see Algorithm 2) or a Gröbner Basis algorithm such as F_4 or F_5 [7].

The XL Algorithm (”eXtended Linearization”) was proposed by Courtois et al. in [4]. In order to solve a quadratic system F, the algorithm enlarges the system by multiplying the polynomials $f^{(i)} \in F$ by all monomials of degree $d \leq D-2$. By doing so, it obtains a large system $\tilde{F}$ of degree D polynomials. The algorithm performs Gaussian Elimination on the system $\tilde{F}$ in order to find a univariate polynomial which then can be solved by Berlekamp's algorithm. In this case, it substitutes the solution into $\tilde{F}$ to simplify the system.
However, if the degree D chosen in step 2 of the algorithm is too small, the enlarged system $\tilde{F}$ will not contain a univariate polynomial. In this case one has to increase the degree D and try again. The smallest degree for which the XL algorithm outputs a solution of the system F is called the *degree of regularity* d_{reg}. This degree of regularity mainly determines the complexity of the algorithm.

For our purposes we want the quadratic system F to be efficiently solvable. In the following, we therefore only consider multivariate systems which can be solved by the XL algorithm at degree 3 or 4. This directly yields an upper bound on the number of variables in our systems. However, for multivariate quadratic systems, it is a hard task to find explicit formulas for this upper bound. In Section 6, we therefore try to find

Algorithm 2 XL-Algorithm

```
Input: Set of polynomials $F = (f^{(1)}, \dots, f^{(m)})$
Output: vector $\mathbf{x} = (x_1, \dots, x_n)$ such that $f^{(1)}(\mathbf{x}) = \dots = f^{(m)}(\mathbf{x}) = 0$
1: for $i = 1$ to $n$ do
2:     Fix an integer $D > 2$.
3:     Generate all polynomials $h \cdot f^{(j)}$ with $h \in T^n_{D-2}$ and $j = 1, \dots, m$.
4:     Perform Gaussian elimination on the set of all polynomials generated in the previous step to generate one equation containing only $x_i$.
5:     If step 4 produced at least one univariate polynomial in $x_i$, solve this polynomial by e.g. Berlekamp's algorithm.
6:     Simplify the equations by substituting the value of $x_i$.
7: end for
8: return $\mathbf{x} = (x_1, \dots, x_n)$
```

these upper bound for concrete instances of Rainbow and HFEv- using a large number of experiments.

5 Reducing the signature size of multivariate schemes

In this section we present our technique to reduce the signature size of multivariate schemes. Our technique can be applied to nearly all multivariate signature schemes, including UOV [11], Rainbow [6], HFEv- [15] and TTS [21].
However, it is not possible to apply our technique directly to more advanced multivariate signature schemes such as QUARTZ and Gui [17]. In order to reduce the signature size of these schemes, we have to modify our technique slightly (see Section 7).

Let $((\mathcal{S}, \mathcal{F}, \mathcal{T}), \mathcal{P})$ be a key pair of a multivariate signature scheme.

Signature Generation: The sender of a message d uses his private key $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ to compute a signature $\mathbf{z} \in \mathbb{F}^n$ for the document d just as in the case of the standard signature scheme. After that, he removes the last k $\mathbb{F}$-elements from the signature $\mathbf{z}$ to obtain a partial signature $\tilde{\mathbf{z}} \in \mathbb{F}^{n-k}$ and sends $\tilde{\mathbf{z}}$ to the verifier.

Signature Verification: The receiver of a signed message checks if $\tilde{\mathbf{z}}$ is indeed part of a valid signature for the document d. To do this, he computes the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ of the document d, substitutes the elements of the partial signature $\tilde{\mathbf{z}}$ into the public key $\mathcal{P}$ and uses one of the techniques described in the previous section to solve the resulting system $\tilde{\mathcal{P}}(z_{n-k+1}, \dots, z_n) = \mathbf{w}$ of m quadratic equations in k variables. If the system has a solution, the signature $\tilde{\mathbf{z}}$ is accepted, otherwise it is rejected.

Remark: Indeed we do not have to reconstruct the complete signature $\mathbf{z} \in \mathbb{F}^n$ by solving the system $\tilde{\mathcal{P}}(z_{n-k+1}, \dots, z_n) = \mathbf{w}$. Instead of this, it suffices to check whether the system has a solution. If we use the Relinearization technique for the verification, it therefore suffices to bring the linearized system into row-echelon form and check if the last equations (which contain only zero terms on the left) hold. The reason for this is given by Proposition 1.

Algorithms 3 and 4 show the standard verification process for multivariate schemes and our modified one in algorithmic form.

5.1 How to choose the parameter k?

In this section we consider the question how we should choose the number k of $\mathbb{F}$-elements removed from the original signature $\mathbf{z}$. Increasing the number k will lead to shorter signatures but increase the computational effort to check the authenticity of a signature.

If $\frac{(k+1)\cdot(k+2)}{2} - 1 \leq m$, the system $\tilde{\mathcal{P}}(\mathbf{x}) = \mathbf{w}$ from step 3 of Algorithm 4 can be solved by the Relinearization technique (see Section 4.1). This means that we can find a solution in polynomial time. Therefore we get

Proposition 1. *Let $\frac{(k+1)\cdot(k+2)}{2} - 1 \leq m$. Then the partial signature $\tilde{\mathbf{z}} \in \mathbb{F}^{n-k}$ can not be found significantly faster than the full signature $\mathbf{z} \in \mathbb{F}^n$.*

Proof. Let us assume that we have a valid partial signature $\tilde{\mathbf{z}} \in \mathbb{F}^{n-k}$ for a document d. By substituting the elements of $\tilde{\mathbf{z}}$ into the public key $\mathcal{P}$ we obtain a system $\tilde{\mathcal{P}}$ of m quadratic equations in k variables. Due to our assumption we can solve this system and therefore recover the whole signature $\mathbf{z}$ in polynomial time using the Relinearization technique (see Section 4.1).

Remark: The above proposition states that an attacker who is able to generate a valid partial signature $\tilde{\mathbf{z}} \in \mathbb{F}^{n-k}$ can generate the whole signature $\mathbf{z} \in \mathbb{F}^n$ quasi without additional computational cost. This shows that the security of the underlying signature schemes is not weakened by our modifications.

Algorithm 3 Standard Verification Algorithm for Multivariate Schemes

```
Input: public key $\mathcal{P}$, document $d$, signature $\mathbf{z} \in \mathbb{F}^n$
Output: boolean value TRUE or FALSE
1: $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$
2: $\mathbf{w}' = \mathcal{P}(\mathbf{z})$
3: if $\mathbf{w}' = \mathbf{w}$ then
4:
5:     return TRUE
6: else
7:     return FALSE
8: end if
```

Algorithm 4 Modified Verification Algorithm for Multivariate Schemes

```
Input: public key $\mathcal{P}$, document $d$, partial signature $\tilde{\mathbf{z}} \in \mathbb{F}^{n-k}$
Output: boolean value TRUE or FALSE
1: $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$
2: $\tilde{\mathcal{P}} = \mathcal{P}(\tilde{\mathbf{z}})$
3: if IsConsistent $(\tilde{\mathcal{P}}(\mathbf{x}) = \mathbf{w})$ = TRUE then
4:     return TRUE
5: else
6:     return FALSE
7: end if
```

From equation (4), we can derive the maximal number k of elements by which we can reduce the length of the original signature $\mathbf{z}$ such that the partial signature $\tilde{\mathbf{z}}$ can be verified using the Relinearization technique by

$$k = \lfloor \frac{1}{2} \cdot (\sqrt{9 + 8 \cdot m} - 3) \rfloor. \tag{5}$$

In the following, we slacken this condition a bit. If $\frac{(k+1)\cdot(k+2)}{2} - 1 > m$, we can not solve the system $\tilde{\mathcal{P}}(\mathbf{x}) = \mathbf{w}$ by the Relinearization technique any more and therefore can not recover the whole signature $\mathbf{z}$ in polynomial time. However, if the number k is not too large, we can solve the system $\tilde{\mathcal{P}}(\mathbf{x}) = \mathbf{w}$ by the XL Algorithm at a very low degree (e.g. $d_{\text{reg}} \in \{3, 4\}$). In this case, the computational effort of recovering the whole signature $\mathbf{z}$ is still very small. We therefore come to the conjecture

Conjecture: If the system $\tilde{\mathcal{P}}(z_{n-k+1}, \ldots, z_n) = \mathbf{w}$ of m quadratic equations in k variables can be solved by the XL Algorithm at degree 3 or 4, our technique does not weaken the security of the underlying signature scheme.

Furthermore, in this case, the additional computational effort needed to verify the reduced signature is still acceptable.
However, as already mentioned in Section 4.2, it is a hard task to find explicit upper bounds for the parameter k such that the system $\tilde{\mathcal{P}}(\mathbf{x}) = \mathbf{w}$ can be solved by the XL Algorithm at a given degree. In the next section we try to find, for concrete instances of Rainbow and HFEv-, these upper bounds by performing a large set of experiments.

6 Results

In this section we show, for concrete instances of the multivariate schemes Rainbow and HFEv-, the possible reduction of the signature size enabled by our technique. Furthermore we analyze the efficiency of our technique using a straightforward implementation of the schemes.

Table 1 shows, for the multivariate signature schemes Rainbow and HFEv-, possible choices of the parameter k of our technique and the resulting reduction in terms of the signature size of the schemes.

For every scheme and parameter set, the 7-th column of the table gives the maximal values of the parameter k such that the partial signature $\tilde{\mathbf{z}}$ can be verified by the Relinearization technique, the XL Algorithm with $d_{\text{reg}} = 3$ and the XL Algorithm with $d_{\text{reg}} = 4$ respectively. While the first of these numbers can be computed using formula (5), the values of k corresponding to the XL Algorithm were obtained experimentally. In the 8-th column of the table we give first the length of a standard signature (without reduction). The second number shows the length of the shortest partial signature which can be verified using the Relinearization technique, while the third and fourth numbers show the lengths of the shortest signatures that can be verified using the XL Algorithm at degree 3 and 4 respectively. The 9-th column shows the corresponding reduction factors.

security level	scheme	private key size (kB)	public key size (kB)	hash size (bit)	verification by	k	signature size (bit)	reduction in %
80	Rainbow(GF(2^8)17,13,13)	19.9	25.1	208	standard	0	344	-
					Relinearization	5	304	12
					XL ($d_{\text{reg}} = 3$)	10	264	23
					XL ($d_{\text{reg}} = 4$)	14	232	33
	HFEv-(GF(7),62,8,2,2)	2.9	47.1	168	standard	0	192	-
					Relinearization	9	165	14
					XL ($d_{\text{reg}} = 3$)	17	135	27
					XL ($d_{\text{reg}} = 4$)	23	126	34
100	Rainbow(GF(2^8),26,16,17)	44.4	59.0	264	standard	0	472	-
					Relinearization	6	424	10
					XL ($d_{\text{reg}} = 3$)	12	384	19
					XL ($d_{\text{reg}} = 4$)	16	344	27
	HFEv-(GF(7),78,8,3,3)	4.5	93.5	210	standard	0	243	-
					Relinearization	11	216	14
					XL ($d_{\text{reg}} = 3$)	17	192	22
					XL ($d_{\text{reg}} = 4$)	24	171	30

Table 1. Possible Reduction of the Signature Length for Rainbow and HFEv-

In the case of the HFEv- signature scheme, we use GF(7) as the underlying field. We store one element of GF(7) in 3 bits, while 5 elements of GF(7) can be efficiently used to store 14 bits. To store a hash value of length 160 bit, we therefore need 60 elements of GF(7), to store a hash value of 200 bit, we need 75 GF(7) elements.
As can be seen from Table 1 we can, in the case of HFEv-, get signatures which are smaller than the input size of the scheme, even when restricting to verifying the partial signatures with the Relinearization technique. But also for the Rainbow scheme, our technique enables us to reduce the signature length by up to 12 %.
When verifying the reduced signatures with the XL Algorithm (with $d_{\text{reg}} = 3$), we can obtain a reduction of the signature length by 20-25 %. When allowing the XL Algorithm to reach degree 4, we can achieve reductions of up to 34 %.

6.1 Efficiency of the Verification Process

To estimate the efficiency of the modified verification process, we created a straightforward implementation of the Rainbow and HFEv- signature schemes in MAGMA code. Our scheme runs on a single core of a server with 24 AMD Opteron processors (2.4 GHz) and 128 GB of RAM. Table 2 shows the time needed for the verification of a (partial) signature (average time of 10,000 verification processes).

As the table shows, there is no significant difference between the running times of the standard verification process and the modified verification process combined with the Relinearization technique. We therefore can achieve a reduction of the signature length by up to 15 % at quasi no cost. A further reduction of the signature length is possible if we accept an increase in the verification time. We hence observe a trade off between signature size and efficiency of the verification process, which, to our knowledge, is unique for multivariate signature schemes and can not be observed for any other class of digital

security level (bit)	scheme	verification by	k	signature size (bit)	reduction in %	verification time (ms)
80	Rainbow(GF(2^8),17,13,13)	standard	0	344	-	0.21
		Relinearization	5	304	12	0.25
		XL Algorithm ($d_{\text{reg}} = 3$)	10	264	23	10.8
		XL Algorithm ($d_{\text{reg}} = 4$)	14	232	33	425.0
	HFEv-(GF(7),62,8,2,2)	standard	0	192	-	0.86
		Relinearization	9	165	14	0.91
		XL Algorithm ($d_{\text{reg}} = 3$)	17	141	27	21.3
		XL Algorithm ($d_{\text{reg}} = 4$)	22	126	34	634.6
100	Rainbow(GF(2^8),26,16,17)	standard	0	472	-	0.42
		Relinearization	6	424	10	0.47
		XL Algorithm ($d_{\text{reg}} = 3$)	12	376	20	14.3
		XL Algorithm ($d_{\text{reg}} = 4$)	16	344	27	534.6
	HFEv-(GF(7),78,8,3,3)	standard	0	243	-	1.25
		Relinearization	10	213	12	1.32
		XL Algorithm ($d_{\text{reg}} = 3$)	19	186	23	37.2
		XL Algorithm ($d_{\text{reg}} = 4$)	24	171	30	928.6

Table 2. Verification times for Rainbow and HFEv- schemes with reduced signature length

signature schemes.

Figure 3 shows this trade off for the example of the Rainbow signature scheme with parameters $(v_1, o_1, o_2) = (17, 13, 13)$. In particular, we note that we can achieve a reduction of the signature size of Rainbow by 5 byte (12 %) at quasi no cost. However, for larger values of k (i.e. solution of $\widetilde{\mathcal{P}}(\mathbf{x}) = \mathbf{w}$ by the XL Algorithm with $d_{\text{reg}} \in \{3, 4\}$) the running time of the verification process increases significantly. More experimental data regarding this trade off between signature size and verification time can be found in Table 2.

7 Application of our Technique to Gui

The Gui signature scheme as proposed by Petzoldt et al. in [17] is currently the multivariate signature scheme with the shortest signatures. In this section we show how to apply our technique to Gui, by which we obtain the shortest signatures of all currently existing signature schemes. However, due to the special signature generation process of Gui, this can not be done straightforward. In order to show this, we start with a short description of Gui.

The Gui signature scheme [17] is an extension of the HFEv- signature scheme introduced in Section 3.2. Indeed, the public and private keys of Gui are just HFEv- keys over GF(2) with specially chosen parameters n, D, a and v. The signature generation process of Gui is very fast [17] and produces very short signatures of size not more than 120 bit. However, due to the parameter choice of Gui, the input length of the HFEv- scheme is only 90 bit. Therefore, it would be possible for an attacker to come up with two messages d_1 and d_2 whose hash values collide in these first 90 bits (birthday attack). To overcome this problem, the authors of [17] developed a specially designed signature generation process for their scheme. Roughly spoken, to generate a signature for a message d, Gui computes

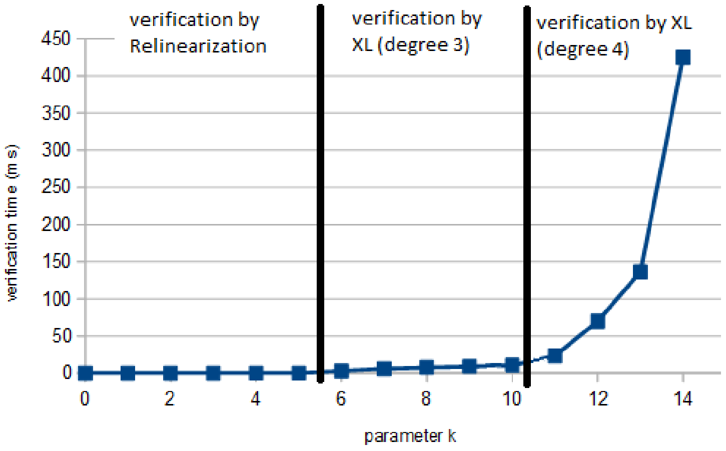

Fig. 3. Verification time (ms) for Rainbow(17,13,13) and different values of k

$r \in \{3,4\}$ HFEv- signatures for different hash values of the document d and combines them to a single Gui signature of length $(n-a)+r\cdot(a+v)$ bits. Algorithm 5 shows the signature generation process of Gui in algorithmic form.
In order to verify a Gui signature $\sigma \in \mathrm{GF}(2)^{(n-a)+r\cdot(a+v)}$, we have to evaluate the public key of Gui r times (see Algorithm 6).

This repeated evaluation of the public key prevents us from applying our technique to Gui directly. However, we are still able to reduce the signature size of Gui by $a+v$ bit.
In particular, while the original Gui signature σ is of the form $\sigma = (S_r, X_r, \ldots, X_1)$, we just transmit the partial signature $\tilde{\sigma} = (S_r, X_r, \ldots, X_2)$. The modified verification algorithm of Gui works as shown in Algorithm 7.
In our modified verification process, the hash values $D_1, \ldots, D_r$ are computed just as in the case of the original Gui scheme (line 3 to 6). We follow the original verification process of Gui (see Algorithm 6) further to compute the vectors $S_{r-1}, \ldots, S_1$ (line 7 to 9). After that, we substitute S_1 into the public key $\mathcal{P}$ (line 10) to obtain a system $\tilde{\mathcal{P}}$ of $n-a$ quadratic equations in $a+v$ variables. The signature $\tilde{\sigma}$ is accepted, if and only if the system $\tilde{\mathcal{P}}(\mathbf{x}) = D_1$ has a solution. For the parameters proposed in [17], this step can be performed by the Relinearization technique (see Section 4.1) in polynomial time.
By our technique, we can therefore reduce the signature size of Gui from $(n-a)+r\cdot(a+v)$ bit to $(n-a)+(r-1)\cdot(a+v)$ bit. Table 3 shows the possible reduction of the signature size for the four parameter sets proposed in [17]. As the table shows, we can, for the parameters $(n, D, a, v, r) = (95, 9, 5, 5, 3)$ (80 bit security), obtain signatures of size 110 bit, which are the shortest signatures of all existing schemes (both in the classical and the post quantum world). Note that by our modifications the security of the scheme is not weakened at all (c.f. Proposition 1), and the performance is not reduced significantly.

Algorithm 5 Signature Generation Process of Gui

```
Input: Gui private key $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ message $d$, repetition factor $r$
Output: signature $\sigma \in \mathrm{GF}(2)^{(n-a)+r(a+v)}$
$\mathbf{h} \leftarrow$ SHA-256($d$)
$S_0 \leftarrow \mathbf{0} \in \mathrm{GF}(2)^{n-a}$
for $i = 1$ to $r$ do
    $D_i \leftarrow$ first $n-a$ bits of $\mathbf{h}$
    $(S_i, X_i) \leftarrow$ HFEv$-^{-1}(D_i \oplus S_{i-1})$
    $\mathbf{h} \leftarrow$ SHA-256($\mathbf{h}$)
end for
$\sigma \leftarrow (S_r || X_r || \ldots || X_1)$
return $\sigma$
```

Algorithm 6 Standard Verification Algorithm of Gui

```
Input: Gui public key $\mathcal{P}$, message $d$, repetition factor $r$, signature $\sigma \in \mathrm{GF}(2)^{(n-a)+r(a+v)}$
Output: boolean value TRUE or FALSE
$\mathbf{h} \leftarrow$ SHA-256($d$)
$(S_r, X_r, \ldots, X_1) \leftarrow \sigma$
for $i = 1$ to $r$ do
    $D_i \leftarrow$ first $n-a$ bits of $\mathbf{h}$
    $\mathbf{h} \leftarrow$ SHA-256($\mathbf{h}$)
end for
for $i = r-1$ to $0$ do
    $S_i \leftarrow \mathcal{P}(S_{i+1} || X_{i+1}) \oplus D_{i+1}$
end for
if $S_0 = \mathbf{0}$ then
    return TRUE
else
    return FALSE
end if
```

Algorithm 7 Modified Verification Algorithm of Gui

```
Input: Gui public key $\mathcal{P}$, message $d$, repetition factor $r$, partial signature $\tilde{\sigma} \in \mathrm{GF}(2)^{(n-a)+(r-1)\cdot(a+v)}$
Output: boolean value TRUE or FALSE
$\mathbf{h} \leftarrow$ SHA-256($\mathbf{d}$)
$(S_r, X_{r-1}, \ldots, X_1) \leftarrow \tilde{\sigma}$
for $i = 1$ to $r$ do
    $D_i \leftarrow$ first $n-a$ bits of $\mathbf{h}$
    $\mathbf{h} \leftarrow$ SHA-256($\mathbf{h}$)
end for
for $i = r-1$ to $1$ do
    $S_i \leftarrow \mathcal{P}(S_{i+1} || X_{i+1}) \oplus D_{i+1}$
end for
$\tilde{\mathcal{P}} \leftarrow \mathcal{P}(S_1)$
if IsConsistent($\tilde{\mathcal{P}}(\mathbf{x}) = D_1$) then
    return TRUE
else
    return FALSE
end if
```

8 Discussion

The technique proposed in Section 5 of this paper enables us to reduce the signature size of multivariate schemes in a very flexible way. In particular, as Figure 3 shows, we achieve a reduction of the signature size by 10-15 % at quasi no cost and a trade off between further reduction and the efficiency of the verification process. Again we note that our modifications do not weaken the security of the schemes.

To our knowledge, this possibility to reduce the signature length without extra cost is unique for multivariate signature schemes, and no other family of signature schemes allows something similar. While, for some lattice based signature schemes [9], there also

security level	scheme	private key size (kB)	public key size (kB)	verification by	k	signature size (bit)	reduction in %
80	Gui(GF(2),96,5,6,6,4)	3.1	61.5	standard	0	126	-
				Relinearization	13	113	10
	Gui(GF(2),95,9,5,5,3)	3.0	59.2	standard	0	120	-
				Relinearization	11	109	9
	Gui(GF(2),94,9,4,4,4)	2.9	56.8	standard	0	122	-
				Relinearization	9	113	7
120	Gui(GF(2),127,9,4,6,4)	5.2	139.2	standard	0	163	-
				Relinearization	11	152	7

Table 3. Possible Reduction of the public key size for Gui

exist techniques to reduce the signature size, they are much less flexible than the proposed technique and require significant extra computation.
Our technique can be applied to every standard multivariate signature scheme such as UOV [11], Rainbow [6], HFEv- [15] and TTS [21] and, with some modifications, to more advanced schemes such as QUARTZ and Gui [17], too. Although, for these schemes, we can not reach as high reduction factors as for Rainbow and HFEv-, we obtain, by applying our technique to Gui, the shortest signatures of all existing signature schemes (both in the classical and the post quantum world).
As demonstrated in Section 2 of this paper, our technique is especially suitable in situations where the connection between sender and receiver is slow. In particular, if the original messages are very short (e.g. status messages of a node in a sensor network) and therefore the signatures are a major part of the communication, our technique helps to reduce the communication cost and therefore speeds up the system significantly. Furthermore, if the small sensors report their status to a computationally powerful server, a slightly more complicated verification process as implied by our technique should be no major problem.

9 Conclusion

In this paper we proposed a general technique to reduce the signature size of multivariate schemes. Our technique enables us to decrease the signature size of nearly all multivariate signature schemes such as Rainbow and HFEv- by up to 15 %, without slowing down the verification process of the schemes significantly. We can prove that the security of the underlying scheme is not weakened by this modification. We can achieve a further reduction of the signature size when accepting a slightly more costly verification process. Therefore, we observe a trade off between signature length and the efficiency of the verification process, which, to our knowledge, is unique for multivariate signature schemes. By applying our technique to the Gui signature scheme of [17], we obtain signatures of size only 110 bit (80 bit security), which are the shortest signatures of all existing digital signature schemes (both in the classical and the post quantum world).

Acknowledgments

The first and the second authors are supported by the EU-project PQCRYPTO ICT-645622 and JSPS KAKENHI 15F15350 respectively.

References

1. D.J. Bernstein, J. Buchmann, E. Dahmen (eds.): Post Quantum Cryptography. Springer, 2009.
2. 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.
3. 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.
4. N. Courtois, A. Klimov, J. Patarin, A. Shamir: Efficient Algorithms for solving overdefined systems of multivariate polynomial equations. EUROCRYPT 2000, LNCS vol. 1807, pp. 392 - 407. Springer 2000.
5. J. Ding, J. E. Gower, D. S. Schmidt: Multivariate Public Key Cryptosystems. Springer, 2006.
6. J. Ding, D. S. Schmidt: Rainbow, a new multivariate polynomial signature scheme. ACNS 2005, LNCS vol. 3531, pp. 164-175. Springer 2005.
7. J.C. Faugère: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra 139, pp. 61-88 (1999).
8. M. R. Garey and D. S. Johnson: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company 1979.
9. T. Güneysu, V. Lyubashevski, T. Pöppelmann: Practical Lattice Based Cryptography: A signature scheme for embedded systems. CHES 2012, LNCS 7428, pp. 530 - 547. Springer 2012.
10. D. Kravitz: Digital Signature Algorithm. US patent 5231668 (July 1991).
11. A. Kipnis, L. Patarin, L. Goubin: Unbalanced Oil and Vinegar Schemes. EUROCRYPT 1999, LNCS vol. 1592, pp. 206–222. Springer 1999.
12. T. Matsumoto, H. Imai: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. EUROCRYPT 1988. LNCS vol. 330, pp. 419-453. Springer 1988.
13. D. Goodin: NSA preps quantum-resistant algorithms to head off crypto-apocalypse. http://arstechnica.com/security/2015/08/nsa-preps-quantum-resistant-algorithms-to-head-off-crypto-apocolypse/.
14. National Institute of Standards and Technology: Report on Post Quantum Cryptography. NISTIR draft 8105, `http://csrc.nist.gov/publications/drafts/nistir-8105/nistir_8105_draft.pdf`.
15. J. Patarin: Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. EUROCRYPT, LNCS vol. 1070, pp. 33 - 48. Springer, 1996.
16. J. Patarin, N. Courtois, L. Goubin: Flash, a fast multivariate signature algorithm. CTRSA 2001, LNCS vol. 2020, pp. 298 - 307. Springer, 2001.
17. A. Petzoldt, M.S. Chen, B.Y. Yang, C. Tao, J. Ding: Design Principles for HFEv- based Signature Schemes. ASIACRYPT 2015 - Part I, LNCS vol. 9452, pp. 311 - 334. Springer 2015.
18. 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).

19. P. Shor: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM J. Comput. 26 (5), pp. 1484 - 1509 (1997).
20. C. Tao, A. Diene, S. Tang, J. Ding. Simple matrix scheme for encryption. PQCrypto 2013, LNCS vol. 7932, pp. 231 – 242. Springer, 2013.
21. 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.