Reducing the Key Size of the SRP Encryption Scheme

Dung Hoang Duong[12], Albrecht Petzoldt[1], and Tsuyoshi Takagi[12]

[1] Institute of Mathematics for Industry, Kyushu University,
744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan
{duong,petzoldt,takagi}@imi.kyushu-u.ac.jp
[2] JST, CREST, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan

Abstract. Multivariate Public Key Cryptography (MPKC) is one of the main candidates for secure communication in a post-quantum era. Recently, Yasuda and Sakurai proposed in [24] a new multivariate encryption scheme called SRP, which offers efficient decryption, a small blow up factor between plaintext and ciphertext and resists all known attacks against multivariate schemes. However, similar to other MPKC schemes, the key sizes of SRP are quite large.
In this paper we propose a technique to reduce the key size of the SRP scheme, which enables us to reduce the size of the public key by up to 54%. Furthermore, we can use the additional structure in the public key polynomials to speed up the encryption process of the scheme by up to 50%. We show by experiments that our modifications do not weaken the security of the scheme.

Keywords: Multivariate Cryptography, SRP Encryption Scheme, Key Size Reduction, Efficiency

1 Introduction

Multivariate cryptography is one of the main candidates to guarantee the security of communication in the post-quantum era [1]. 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 RFIDs or smart cards [3,5]. Especially in the area of *signature* schemes, there exist many practical multivariate schemes such as UOV [14], Rainbow [9] and Gui [20]. On the other hand, the development of public key *encryption* schemes based on multivariate polynomials appeared to be a much harder task. Indeed, many multivariate encryption schemes such as MI [16] and HFE [17] have been broken in the past. Currently, there exists very few candidates for encryption schemes including ZHFE [21] and ABC [22]. Therefore, the development of secure and efficient multivariate encryption schemes is an important research topic.

Recently, Yasuda and Sakurai proposed in [24] a new multivariate encryption scheme called SRP, which combines the Square encryption scheme [6], the Rainbow signature scheme [9] and the Plus method [8]; hence the name SRP. Since both the decryption process of Square and the inversion of the Rainbow central map are very efficient, the decryption process of SRP is quite fast. Furthermore, unlike many other multivariate encryption schemes, the blow up factor between plaintext and ciphertext is less than two. In addition, by combining different schemes into one, several attacks against multivariate schemes are not applicable against SRP, cf. [24, Section 4]. However, similar to other multivariate schemes, the sizes of the public and private key of SRP are relatively large.

In this paper we propose a technique to reduce the public key size of the SRP scheme. By our technique it is possible to reduce the size of the SRP public key by up to 54%. Furthermore, it allows to reduce the number of field multiplications needed during the encryption process of the scheme by up to 50%. We show by experiments that the security of the SRP scheme is not weakened by our modifications. While a similar approach to reduce the public key size has been made by Petzoldt et al. [18,19] for multivariate *signature* schemes such as UOV and Rainbow, our technique is the first approach to reduce the public key size of a multivariate *encryption* scheme.

By our modifications, we obtain a very efficient multivariate encryption scheme. The public key size of the scheme is about 50% smaller than that of other multivariate schemes such as ABC and ZHFE. The encryption process is about twice as fast as that of the other schemes. Furthermore, the decryption process of our proposed scheme is as fast as that of the standard SRP scheme.

Our paper is organized as follows. In Section 2, we recall the basic concepts of multivariate public key cryptography and the SRP encryption scheme. Section 3 presents the construction of our CyclicSRP scheme and analyzes the security of our construction. In Section 4 we give concrete parameter sets for our scheme and compare it with the standard SRP scheme with regard to key sizes. Section 5 describes how the additional structure in the public key of our scheme can be used to speed up the encryption process, and Section 6 concludes the paper.

2 The SRP Encryption Scheme

In this section, we recall the basic SRP scheme of [24]. Before we come to the description of the scheme itself, we start with a short overview of the basic concepts of multivariate cryptography.

2.1 Multivariate cryptography

The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials over a finite field $\mathbb{F}$. The security of multivariate schemes is based on the *MQ Problem* of solving such a system. The MQ Problem is proven to be NP-Hard even for quadratic polynomials over the field GF(2) [12].

To build a multivariate public key cryptosystem (MPKC), 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, we compose 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 the three maps $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert the public key. To encrypt a message $M \in \mathbb{F}^n$, one simply computes $C = \mathcal{P}(M) \in \mathbb{F}^m$. To decrypt a ciphertext $C \in \mathbb{F}^m$, one computes recursively $\mathbf{x} = \mathcal{S}^{-1}(C) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$ and $M = \mathcal{T}^{-1}(\mathbf{y})$. $M \in \mathbb{F}^n$ is the plaintext corresponding to the ciphertext C.

Since, for multivariate encryption schemes, we have $m \geq n$, the pre-image of the vector $\mathbf{x}$ under the central map $\mathcal{F}$ and therefore the decrypted plaintext will (with overwhelming probability) be unique.

2.2 SRP

The SRP encryption scheme was recently proposed by Yasuda and Sakurai in [24] by combining the Square encryption scheme [6], the Rainbow signature scheme [9] and the Plus method [8]. By combining Square and Rainbow into one scheme, several attacks against the single schemes are not longer applicable. Furthermore, since both Square and Rainbow are very efficient, the same holds for the SRP scheme.

The combination of Square with a signature scheme has another important benefit: The central map of Square is no one-to-one mapping. The combination with Rainbow provides an efficient possibility to distinguish between correct and false solutions.

For our purpose, we restrict to variants of SRP in which the Rainbow part is replaced by UOV [14]. Note that the parameter sets proposed in [24] are of this type.

We choose a finite field $\mathbb{F} = \mathbb{F}_q$ of odd characteristic with $q \equiv 3 \bmod 4$ and, for an odd integer d, a degree d extension field $\mathbb{E} = \mathbb{F}_{q^d}$. Let $\phi : \mathbb{E} \to \mathbb{F}^d$ be an isomorphism between the field $\mathbb{E}$ and the vector space $\mathbb{F}^d$. Moreover, let o, r, s and l be non-negative integers.

Key Generation Let $n = d + o - l$, $n' = d + o$ and $m = d + o + r + s$. The *central map* $\mathcal{F} : \mathbb{F}^{n'} \to \mathbb{F}^m$ of the scheme is the concatenation of three maps $\mathcal{F}_S$, $\mathcal{F}_R$, and $\mathcal{F}_P$. These maps are defined as follows.

(i) The Square part $\mathcal{F}_S : \mathbb{F}^{n'} \to \mathbb{F}^d$ is the composition of the maps

$$\mathbb{F}^{d+o} \xrightarrow{\pi_d} \mathbb{F}^d \xrightarrow{\phi^{-1}} \mathbb{E} \xrightarrow{X \mapsto X^2} \mathbb{E} \xrightarrow{\phi} \mathbb{F}^d.$$

Here $\pi_d : \mathbb{F}^{d+o} \to \mathbb{F}^d$ is the projection to the first d coordinates.

(ii) The UOV (Rainbow) part $\mathcal{F}_R = (f^{(1)}, \ldots, f^{(o+r)}) : \mathbb{F}^{n'} \to \mathbb{F}^{o+r}$ is constructed as the usual UOV signature scheme: let $V = \{1, \ldots, d\}$ and $O = \{d + 1, \ldots, d + o\}$. For every $k \in \{1, \ldots, o + r\}$, the quadratic polynomial $f^{(k)}$ is of the form

$$f^{(k)}(x_1, \ldots, x_{n'}) = \sum_{i \in O, j \in V} \alpha_{i,j}^{(k)} x_i x_j + \sum_{i,j \in V, i \leq j} \beta_{i,j}^{(k)} x_i x_j + \sum_{i \in V \cup O} \gamma_i^{(k)} x_i + \eta^{(k)},$$

with $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)}$ randomly chosen in $\mathbb{F}$. [3]

(iii) The Plus part $\mathcal{F}_P = (g^{(1)}, \ldots, g^{(s)}) : \mathbb{F}^{n'} \to \mathbb{F}^s$ consists of s randomly chosen quadratic polynomials $g^{(1)}, \ldots, g^{(s)}$.

We additionally choose an affine embedding $\mathcal{T} : \mathbb{F}^n \hookrightarrow \mathbb{F}^{n'}$ of full rank and an affine isomorphism $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$. The *public key* is given by $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$ and the *private key* consists of $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$.

$$\begin{tikzcd}
& & \mathbb{F}^d \arrow[rd] & & \\
\mathbb{F}^n \arrow[r, "\mathcal{T}"] \arrow[rrrr, bend right=60, "\mathcal{P}"'] & \mathbb{F}^{n+l} \arrow[ru, "\mathcal{F}_S"] \arrow[r, "\mathcal{F}_R"] \arrow[rd, "\mathcal{F}_P"'] & \mathbb{F}^{o+r} \arrow[r] & \mathbb{F}^m \arrow[r, "\mathcal{S}"] & \mathbb{F}^m \\
& & \mathbb{F}^s \arrow[ru] & &
\end{tikzcd}$$

Encryption: Given a message $M \in \mathbb{F}^n$, the ciphertext C is computed as $C = \mathcal{P}(M) \in \mathbb{F}^m$.

Decryption: Given a ciphertext $C = (c_1, \ldots, c_m)$, the decryption is executed as follows.

(1) Compute $\mathbf{x} = (x_1, \ldots, x_m) = \mathcal{S}^{-1}(C)$.

(2) Compute $X = \phi^{-1}(x_1, \ldots, x_d)$.

(3) Compute $R_{1,2} = \pm X^{(q^d+1)/4}$ and set $\mathbf{y}^{(i)} = (y_1^{(i)}, \ldots, y_d^{(i)}) = \phi(R_i)$ $(i = 1, 2)$. [4]

(4) Given the vinegar values $y_1^{(i)}, \cdots, y_d^{(i)}$ $(i = 1, 2)$, solve the two systems of $o + r$ linear equations in $n' - d = o$ variables $u_{d+1}, \cdots, u_{n'}$ given by

$$f^{(k)}(y_1^{(i)}, \ldots, y_d^{(i)}, u_{d+1}, \cdots, u_{n'}) = x_{d+k} \quad (i = 1, 2)$$

for $k = 1, \cdots, o + r$. The solution is denoted by $(y_{d+1}, \cdots, y_{n'})$. [5]

(5) Compute the plaintext $M \in \mathbb{F}^n$ by finding the pre-image of $(y_1, \cdots, y_{n'})$ under the affine embedding $\mathcal{T}$.

[3] Note that, while, in the standard UOV signature scheme, we only have o polynomials, the map $\mathcal{F}_R$ consists of $o + r$ polynomials of the Oil and Vinegar type. This fact is needed to reduce the probability of decryption failures (see footnote 3).

[4] The fact of $q = 3 \bmod 4$ and d odd allows us to compute the square roots of X by this simple operation. Therefore, the decryption process of both Square and SRP is very efficient.

[5] In [24, Proposition 1] it was shown that the probability of both $(y_1^{(1)}, \ldots, y_d^{(1)})$ and $(y_1^{(2)}, \ldots, y_d^{(2)})$ leading to a solution of the linear system is about $1/q^{-r-1}$. Therefore, with overwhelming probability, one of the possible solutions is eliminated during this step.

Note that the only part of the central map needed for decryption are the coefficients of the Rainbow polynomials $f^{(1)}, \ldots, f^{(o+r)}$.

In the following, we restrict to a homogeneous quadratic map $\mathcal{F}$ as well as to linear maps $\mathcal{S}$ and $\mathcal{T}$. Therefore, the public key $\mathcal{P}$ of the scheme will be a homogeneous quadratic system, too. The number of terms in each component of the public key is given by $\frac{n \cdot (n+1)}{2} =: D$.

3 Our Improved Scheme

In this section, we present our technique to generate a key pair for SRP with a structured public key. In particular we are able to construct a public key of the form shown in Figure 1.

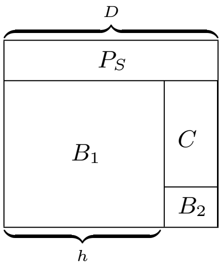

Fig. 1. Structure of the public key $\mathcal{P}$

Here, the matrices $B_1 \in \mathbb{F}^{(m-d) \times h}$ and $B_2 \in \mathbb{F}^{s \times (D-h)}$ can be arbitrarily chosen by the user, and the parameter h is given by $h = \frac{d \cdot (d+1)}{2} + d \cdot (o - l)$.

In the following, we choose the matrices B_1 and B_2 in a "cyclic" way. In particular, we choose two random vectors $\mathbf{b}_1 \in \mathbb{F}^h$ and $\mathbf{b}_2 \in \mathbb{F}^{D-h}$. The first row of the matrix B_1 is just the vector $\mathbf{b}_1$, while the i-th row of B_1 corresponds to a cyclic right shift of the vector $\mathbf{b}_1$ by $i-1$ positions ($i = 2, \ldots, m-d$). Analogously, the first row of the matrix B_2 corresponds to the vector $\mathbf{b}_2$ and the i-th row of this matrix is the cyclic right shift of $\mathbf{b}_2$ by $i-1$ positions.
By choosing the matrices B_1 and B_2 in this way, we have to store only the two vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ to recover the matrices B_1 and B_2. Therefore, the public key size of the scheme is reduced significantly (see Section 4). Furthermore, we can use the structure in the matrices B_1 and B_2 to speed up the encryption process of the scheme (see Section 5). The resulting scheme is called CyclicSRP.
Appendix A of this paper shows an algorithm for creating B_1 and B_2 in this way. Furthermore, we discuss there shortly two other possibilities of constructing B_1 and B_2.

3.1 Notations

Let $\mathcal{P} = (p^{(1)}, \cdots, p^{(m)})$, $\mathcal{F} = (f^{(1)}, \cdots, f^{(m)})$ and $\mathcal{Q} = \mathcal{F} \circ \mathcal{T} = (q^{(1)}, \cdots, q^{(m)})$ with

$$\begin{aligned} f^{(k)}(x_1, \cdots, x_{n'}) &= \sum_{1 \le i \le j \le n'} f_{ij}^{(k)} x_i x_j, \\ q^{(k)}(x_1, \cdots, x_n) &= \sum_{1 \le i \le j \le n} q_{ij}^{(k)} x_i x_j, \\ p^{(k)}(x_1, \cdots, x_n) &= \sum_{1 \le i \le j \le n} p_{ij}^{(k)} x_i x_j \end{aligned}$$

for each $k = 1, \ldots, m$. These coefficients are written into matrices F, Q and P according to the lexicographic order. Note here again that we only consider homogeneous quadratic maps $\mathcal{F}$, $\mathcal{Q}$ and $\mathcal{P}$. [6]
Let $S = (s_{ij})_{1 \le i \le m}^{1 \le j \le m}$ and $T = (t_{ij})_{1 \le i \le n'}^{1 \le j \le n}$ be the matrix representations of the linear maps $\mathcal{S}$ and $\mathcal{T}$ respectively.
Furthermore, we divide the matrices S, Q and F into submatrices as shown in Figure 2.

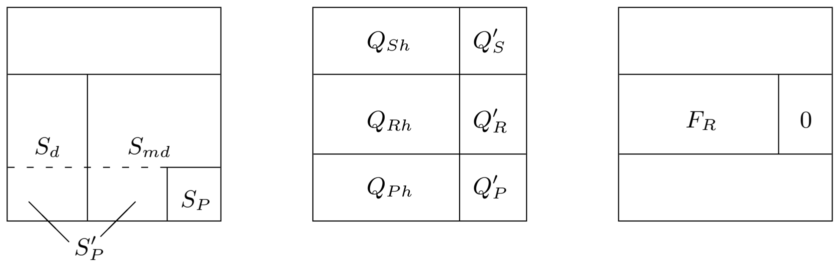

Fig. 2. Layout of the matrices S, Q and F

Additionally, we define $Q_S = (Q_{Sh} \| Q'_S) \in \mathbb{F}^{d \times D}$ and $Q_{RPh} = \begin{pmatrix} Q_{Rh} \\ Q_{Ph} \end{pmatrix} \in \mathbb{F}^{(o+r+s) \times h}$.

3.2 Construction

After fixing the matrices S, T, B_1 and B_2, the entries of the matrix Q_S (i.e. the coefficients of the map $\mathcal{Q}$ referring to the Square part of SRP) are determined by the equation

$$Q_S(\mathbf{x}) = \phi\left((\phi^{-1} \circ \pi_d \circ \mathcal{T}(\mathbf{x}))^2\right) = (q^{(1)}(\mathbf{x}), \ldots, q^{(d)}(\mathbf{x})). \tag{1}$$

[6] Due to the embedding function $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^{n'}$, the number of variables in $\mathcal{F}$ is given by n', while $\mathcal{Q}$ and $\mathcal{P}$ contain only n variables.

From $\mathcal{P} = \mathcal{S} \circ \mathcal{Q}$ it follows directly that $P = S \cdot Q$. Therefore we obtain $B_1 = S_d \cdot Q_{Sh} + S_{md} \cdot Q_{RPh}$ which, under the assumption of S_{md} being invertible, yields

$$Q_{RPh} = S_{md}^{-1} \cdot (B_1 - S_d \cdot Q_{Sh}). \tag{2}$$

Furthermore, from $\mathcal{Q} = \mathcal{F} \circ \mathcal{T}$ we obtain the relation

$$q_{ij}^{(k)} = \sum_{r=1}^{n'} \sum_{s=r}^{n'} \alpha_{ij}^{rs} f_{rs}^{(k)} \quad (1 \leq i \leq j \leq n) \tag{3}$$

for each $k = 1, \ldots, m$, where

$$\alpha_{ij}^{rs} = \begin{cases} t_{ri} t_{si} & \text{if i = j} \\ t_{ri} t_{sj} + t_{rj} t_{si} & \text{otherwise.} \end{cases}$$

We consider the $m - d - s = o + r$ equations from (3) for $k = d + 1, \ldots, m - s$; those correspond to the UOV part of SRP. Due to the special structure of the UOV polynomials, we have

$$q_{ij}^{(k)} = \sum_{r=1}^{d} \sum_{s=r}^{n'} \alpha_{ij}^{rs} f_{rs}^{(k)} \quad (1 \leq i \leq j \leq n,\ d + 1 \leq k \leq m - s). \tag{4}$$

Let A be the $(d(d+1)/2 + od) \times h$ matrix containing the coefficients α_{ij}^{rs} of equation (4) for $1 \leq r \leq d, r \leq s \leq n'$ for the rows and $1 \leq i \leq d,\ i \leq j \leq n$ for the columns. With this notation, equation (4) yields

$$Q_{Rh} = F_R \cdot A. \tag{5}$$

If A has full rank, we therefore can recover F_R from Q_{Rh} by solving, for each $k \in \{d + 1, \ldots, m - s\}$, a linear system of the form

$$\begin{pmatrix} q_{11}^{(k)} \\ q_{12}^{(k)} \\ \vdots \\ q_{dn}^{(k)} \end{pmatrix} = A^T \cdot \begin{pmatrix} f_{11}^{(k)} \\ f_{12}^{(k)} \\ \vdots \\ f_{dn'}^{(k)} \end{pmatrix}. \tag{6}$$

Remark: 1) Experiments show that, for a randomly chosen invertible matrix T, the probability of A having rank h is quite high. Therefore, we do not have to test many matrices T to find a matrix A of full rank.
2) The linear systems in equation (6) have multiple solutions. We just randomly choose one of these solutions and put it into the matrix F_R.

Having recovered the coefficients of the Rainbow central map, we can easily compute the elements of the matrix Q'_R by using the relation $\mathcal{Q} = \mathcal{F} \circ \mathcal{T}$.

The last submatrix of Q still unknown is now Q'_P. Under the assumption of S_P being invertible we can recover it by

$$Q'_P = S_P^{-1} \cdot \left(B_2 - S'_P \cdot \begin{pmatrix} Q'_S \\ Q'_R \end{pmatrix} \right). \tag{7}$$

Having therefore recovered the whole matrix Q, it is easy to compute the coefficient matrix of the public key by

$$P = S \cdot Q. \tag{8}$$

Note that the so computed matrix P will have the structure shown in Figure 1.

We publish P as the public key of our scheme, while the private key consists of S, T and F_R.

Algorithm 1 shows this key generation process in compact form.

Algorithm 1 Key Generation of our SRP variant

```
Input: SRP parameters $q, d, o, r, s, l$, matrices $B_1 \in \mathbb{F}^{(m-d)\times h}$ and $B_2 \in \mathbb{F}^{s\times(D-h)}$.
Output: SRP key pair $((S, F_R, T), P)$ with $P$ of the form of Figure 1.
Choose an invertible matrix $S \in \mathbb{F}^{m\times m}$ such that the submatrices
    $S_{md} \in \mathbb{F}^{(m-s)\times(m-s)}$ and $S_p \in \mathbb{F}^{s\times s}$ are invertible.
Choose a full rank matrix $T \in \mathbb{F}^{n'\times n}$ such that the matrix $A$ has full rank.
Compute $Q_S$ by equation (1).
Compute $Q_{RPh}$ by equation (2).
Compute $F_R$ by equation (6).
Compute $Q_R$ using the relation $\mathcal{Q} = \mathcal{F} \circ \mathcal{T}$.
Compute $Q'_P$ by equation (7).
Compute $P = S \cdot Q$.
return $((S, F_R, T), P)$
```

3.3 Security

The security analysis of our scheme runs in the same way as for the standard SRP scheme of [24]. We therefore refer to [24, Section 4] regarding an analysis of our scheme against rank attacks [2,13], and attacks of the UOV type [15,14,10], and only cover here direct attacks [1,23].

Direct Attacks [1,23] The direct attack tries to recover the plaintext M by solving the public system $\mathcal{P}(M) = C$ as an instance of the MQ Problem using an algorithm like XL or a Gröbner Basis method.
To study the security of the CyclicSRP scheme against direct attacks, we carried out a large number of experiments with MAGMA [4] ver. 2.18-9, which contains

an efficient implementation of Faugéres F_4-algorithm [11] for computing Gröbner Bases. Table 1 shows the results of our experiments against random systems, the SRP scheme and our scheme. For each parameter set, we carried out 10 experiments.

Table 1. Results of experiments with direct attacks

parameters		CyclicSRP		SRP		random system	
q,d,o,r,s,l	m,n	d_{reg}	time (s)	d_{reg}	time (s)	d_{reg}	time (s)
$31,11,10,5,4,6$	$30,15$	4	3.2	4	3.2	4	3.2
$31,11,10,5,4,5$	$30,16$	5	31.8	5	31.8	5	32.0
$31,11,10,5,4,4$	$30,17$	5	91.8	5	92.7	5	94.0
$31,11,10,5,4,3$	$30,18$	5	382.8	5	382.2	5	470.3
$31,11,10,5,4,2$	$30,19$	6	$4,646$	6	$4,650$	6	$5,785$

As the table shows, the F_4 algorithm can not solve our systems significantly faster than those of the standard SRP scheme.

4 Parameters and Key Sizes

4.1 Parameters

In this paper, we consider three parameter sets proposed by Yasuda in [24]. In particular, these are

(A) $(q,d,o,r,s,l) = (31,33,32,16,5,16)$ providing 80 bit of security
(B) $(q,d,o,r,s,l) = (31,47,47,22,5,22)$ providing 112 bit of security and
(C) $(q,d,o,r,s,l) = (31,71,71,32,5,32)$ proving a security level of 160 bit.

4.2 Key Size

The size of the public key of the standard SRP scheme [24] is

$$m \cdot \frac{n(n+1)}{2} \text{ field elements.} \tag{9}$$

Note here again that we restrict to a homogeneous quadratic public map.

The public key of the CyclicSRP scheme consists of the matrix P_S representing the Square part, the vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ and the matrix C (see Figure 1).

Therefore, the size of the public key is given by

$$\begin{aligned} \text{size}_{\text{pk}} &= d \cdot D + h + D - h + (m - d - s) \cdot (D - h) \\ &= (m + 1 - s) \cdot \frac{n(n+1)}{2} - (m - d - s) \cdot \left(\frac{d(d+1)}{2} + d(o - l) \right) \end{aligned} \tag{10}$$

field elements. Table 2 gives a comparison between the standard SRP scheme and our scheme with the three parameter sets from [24].

Table 2. Public key size comparison of SRP scheme and our scheme

		(A)	(B)	(C)
Parameters	q, d, o, r, s, l	$31, 33, 32, 16, 5, 16$	$31, 47, 47, 22, 5, 22$	$31, 71, 71, 32, 5, 32$
	m, n	$86, 49$	$121, 72$	$179, 110$
Public key size	Standard SRP	$105, 350$	$317, 988$	$1, 092, 795$
	CyclicSRP	$48, 178$	$148, 569$	519.900
	Reduction	54.3%	53.3%	52.4%

5 Efficiency of the encryption process

Besides the considerable reduction of the public key size, we can use the additional structure in the public key of CyclicSRP to reduce the the number of multiplications needed in the encryption process significantly. This can be seen as follows.

The encryption process of a multivariate encryption scheme consists of the evaluation of the public system $\mathcal{P}$. Basically, there are two approaches to do this.

In the first approach we store the public key $\mathcal{P}$ of the scheme as a matrix $P \in \mathbb{F}^{m \times n(n+1)/2}$. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}^n$ and define $\mathbf{X} = (x_1^2, x_1 x_2, \cdots, x_n^2) \in \mathbb{F}^{n(n+1)/2}$ to be the vector containing the values of the quadratic monomials of $\mathbb{F}[x_1, \ldots, x_n]$ in lexicographical order. Then we have

$$\mathcal{P}(\mathbf{x}) = P \cdot \mathbf{X}^T. \tag{11}$$

To evaluate the public key $\mathcal{P}$ using this approach, we need

- $n(n+1)/2$ multiplications to compute the vector $\mathbf{X}$ and
- $m \cdot n(n+1)/2$ multiplications to compute the matrix vector product $P \cdot \mathbf{X}^T$.

Altogether, this approach requires

$$(m+1) \cdot n(n+1)/2 \;\; \text{field multiplications} \tag{12}$$

to evaluate the system $\mathcal{P}$.

For the second approach, we store the public key in the form of m upper triangular matrices $P^{(i)} \in \mathbb{F}^{n \times n}$ of the form

$$P^{(i)} = \begin{pmatrix} p_{11}^{(i)} & p_{12}^{(i)} & \cdots & p_{1n}^{(i)} \\ 0 & p_{22}^{(i)} & \cdots & p_{2n}^{(i)} \\ 0 & 0 & \vdots & \ddots \\ 0 & 0 & \cdots & p_{nn}^{(i)} \end{pmatrix} \quad (i = 1, \ldots, m).$$

Let $\mathbf{x} = (x_1, \ldots, x_n)$. Then we have

$$\mathcal{P}(\mathbf{x}) = (\mathbf{x} \cdot P^{(1)} \cdot \mathbf{x}^T, \ldots, \mathbf{x} \cdot P^{(m)} \cdot \mathbf{x}^T)^T. \tag{13}$$

To evaluate one random polynomial in this way, we need

- $n(n+1)/2$ multiplications to compute $\mathbf{x} \cdot P^{(i)}$ and
- n multiplications to compute the inner product $(\mathbf{x} \cdot P^{(i)}) \cdot \mathbf{x}^T$.

Hence, in order to evaluate the public key $\mathcal{P}$ using the second approach, we need

$$m \cdot \frac{n \cdot (n+3)}{2} \text{ field multiplications.} \tag{14}$$

However, when evaluating the public system of our CyclicSRP scheme using this approach, we do not have to perform all these multiplications one by one.
To be more specific, for the first d polynomials corresponding to the Square part, the needed number of multiplications is computed as above; we need

$$d \cdot \frac{n \cdot (n+3)}{2} \text{ field multiplications.}$$

The first polynomial $p^{(d+1)}$ belonging to the Rainbow part is also evaluated in the standard way.
However, when we look at the matrix $P^{(d+2)}$ containing the coefficients of the second polynomial of the Rainbow part , we see that many of the computations we need in order to compute $\mathbf{x} \cdot P^{(d+2)}$ have already been performed during the evaluation of $p^{(d+1)}$ (see equation (15)). In particular, for the example shown in equation (15), these are the computations $a \cdot x_1$, $b \cdot x_1 + f \cdot x_2$, $c \cdot x_1 + g \cdot x_2 + j \cdot x_3$ and $d \cdot x_1 + h \cdot x_2 + k \cdot x_3$. By systematically reusing these results, we can therefore save a large number of multiplications.

$$P^{(d+1)} = \left(\begin{array}{ccccc} a & b & c & d & e \\ 0 & f & g & h & i \\ 0 & 0 & j & k & l \\ \hline 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{array}\right) \qquad P^{(d+2)} = \left(\begin{array}{ccccc} l & a & b & c & d \\ 0 & e & f & g & h \\ 0 & 0 & i & j & k \\ \hline 0 & 0 & 0 & * & * \\ 0 & 0 & 0 & 0 & * \end{array}\right) \tag{15}$$

During the evaluation of the Plus polynomials $p^{(d+o+r+1)}, \ldots, p^{(m)}$, the number of reusable results is even higher.

As a thorough analysis shows, we can, by using this strategy, save

$$\begin{aligned} &(o+r-1) \cdot \sum_{i=1}^{d} (n-i) + (s-1) \cdot \sum_{i=1}^{n} (n-i) \\ = &(o+r+s-2) \cdot \frac{n \cdot (n-1)}{2} - (o+r-1) \cdot \frac{(n-d) \cdot (n-d-1)}{2} \end{aligned} \tag{16}$$

field multiplications. Table 3 gives a comparison of the number of field multiplications needed during the encryption process of the standard SRP scheme and CyclicSRP for the three parameter sets of [24]. The two numbers given for the standard SRP scheme refer to the evaluation of the public system with the first and the second approach respectively. The numbers in the last row give the ratio of saved field multiplications between CyclicSRP and the standard SRP scheme (when evaluating the polynomials with the first approach).

Remark: When evaluating the polynomials $p^{(1)}, \ldots, p^{(d)}$ as well as the matrix C with the first approach, we achieve a further reduction of the number of field multiplications needed during the encryption process of

$$d \cdot n - \frac{n+1}{2} + (o+r) \cdot \left(\frac{(n-d) \cdot (n-d+3)}{2} - \frac{(o-l) \cdot (o-l+1)}{2} \right). \quad (17)$$

The numbers given in Table 3 refer to this strategy.

Table 3. Comparison between numbers of multiplications needed in the encryption process of SRP and CyclicSRP

		(A)	(B)	(C)
Parameters	q, d, o, r, s, l	$31, 33, 32, 16, 5, 16$	$31, 47, 47, 22, 5, 22$	$31, 71, 71, 32, 5, 32$
	m, n	$86, 49$	$121, 72$	$179, 110$
# field multiplications during encryption	Standard SRP	106,575	320,616	1,098,900
		109,564	326,700	1,112,485
	CyclicSRP	54,068	160,587	546,875
	Reduction	49.3%	49.9%	50.2%

6 Conclusion

In this paper we investigated the recent multivariate encryption scheme SRP [24] which is a good candidate for post-quantum encryption schemes. We proposed a technique to reduce the public key size of this scheme. The resulting scheme, CyclicSRP, reduces the size of the public key by up to 54% and the number of field multiplications needed during the encryption process by 50%. By our technique we therefore help to solve one of the biggest problems of multivariate schemes, namely the large size of the public keys. To our knowledge, our proposal is the first application of such a technique to a multivariate encryption scheme. Future work includes

- Application of our technique to other multivariate encryption schemes such as ABC.
- Extension of our technique to SRP variants with several Rainbow layers.

References

1. Bernstein, D. J., Buchmann, J., and Dahmen, E., Eds. *Post-quantum cryptography.* Springer-Verlag, 2009.
2. Billet, O., and Gilbert, H. Cryptanalysis of Rainbow. In *Security and Cryptography for Networks—SCN 2006*, vol. 4116 of *Lecture Notes in Comput. Sci.* Springer, 2006, pp. 336–347.
3. Bogdanov, A., Eisenbarth, T., Rupp, A., and Wolf, C. Time-Area optimized public-key engines: MQ-Cryptosystems as replacement for elliptic curves? In *Cryptographic Hardware and Embedded Systems—CHES 2008*, vol. 5154 of *Lecture Notes in Comput. Sci.* Springer, 2008, pp. 45–61.
4. Bosma, W., Cannon, J., and Playoust, C. The Magma algebra system. I. The user language. *J. Symbolic Comput. 24*, 3-4 (1997), 235–265.
5. Chen, A. I.-T., Chen, M.-S., Chen, T.-R., Cheng, C.-M., Ding, J., Kuo, E. L.-H., Lee, F. Y.-S., and Yang, B.-Y. SSE implementation of Multivariate PKCs on modern x86 CPUs. In *Cryptographic Hardware and Embedded Systems—CHES 2008*, vol. 5154 of *Lecture Notes in Comput. Sci.* Springer, 2008, pp. 33–48.
6. Clough, C., Baena, J., Ding, J., Yang, B.-Y., and shing Chen, M. Square, a new multivariate encryption scheme. In *Topics in Cryptology—CT-RSA 2009*, vol. 5473 of *Lecture Notes in Comput. Sci.* Springer, 2009, pp. 252–264.
7. Ding, J., Clough, C., and Araujo, R. Inverting square systems algebraically is exponential. *Finite Fields and Their Applications 26* (2014), 32–48.
8. Ding, J., Gower, J. E., and Schmidt, D. S. *Multivariate public key cryptosystems*, vol. 25 of *Advances in Information Security.* Springer, 2006.
9. Ding, J., and Schmidt, D. Rainbow, a new multivariable polynomial signature scheme. In *Applied Cryptography and Network Security, Third International Conference, ACNS 2005*, vol. 3531 of *Lecture Notes in Comput. Sci.*, Springer, pp. 164–175.
10. Ding, J., Yang, B.-Y., Chen, C.-H. O., Chen, M.-S., and Cheng, C.-M. New Differential-Algebraic attacks and reparametrization of Rainbow. In *Applied Cryptography and Network Security—ACNS 2008*, vol. 5037 of *Lecture Notes in Comput. Sci.* Springer, 2008, pp. 242–257.
11. Faugére, J.-C. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra 139*, 1-3 (1999), 61–88.
12. Garey, M. R., and Johnson, D. S. *Computers and intractability: A guide to the theory of NP-completeness.* W. H. Freeman, 1979.
13. Goubin, L., and Courtois, N. T. Cryptanalysis of the TTM cryptosystem. In *Advances in Cryptology—ASIACRYPT 2000*, vol. 1976 of *Lecture Notes in Comput. Sci.* Springer, 2000, pp. 44–57.
14. Kipnis, A., Patarin, J., and Goubin, L. Unbalanced oil and vinegar signature schemes. In *EUROCRYPT 1999*, vol. 1592 of *Lecture Notes in Comput. Sci.* Springer, 1999, pp. 206–222.
15. Kipnis, A., and Shamir, A. Cryptanalysis of the oil and vinegar signature scheme. In *Advances in cryptology—CRYPTO '98*, vol. 1462 of *Lecture Notes in Comput. Sci.* Springer, 1998, pp. 257–266.
16. Matsumoto, T., and Imai, H. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Advances in cryptology—EUROCRYPT '88*, vol. 330 of *Lecture Notes in Comput. Sci.* Springer, 1988, pp. 419–453.

17. Patarin, J. Hidden field equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In *Advances in cryptology—EUROCRYPT*, vol. 1070 of *Lecture Notes in Comput. Sci.* Springer, 1996, pp. 33–48.
18. Petzoldt, A., Bulygin, S., and Buchmann, J. CyclicRainbow—a multivariate signature scheme with a partially cyclic public key. In *Progress in cryptology—INDOCRYPT 2010*, vol. 6498 of *Lecture Notes in Comput. Sci.* Springer, 2010, pp. 33–48.
19. Petzoldt, A., Bulygin, S., and Buchmann, J. A. Linear recurring sequences for the UOV key generation. In *Public Key Cryptography - PKC 2011 - 14th International Conference on Practice and Theory in Public Key Cryptography* (2011), pp. 335–350.
20. Petzoldt, A., Chen, M.-S., Yang, B.-Y., Tao, C., and Ding, J. Design principles for hfev- based multivariate signature schemes. In *ASIACRYPT 2015*, vol. 9742 of *Lecture Notes in Comput. Sci.* Springer, 2015, pp. 1–24.
21. Porras, J., Baena, J., and Ding, J. ZHFE, a new multivariate public key encryption scheme. In *Post-quantum cryptography*, vol. 8772 of *Lecture Notes in Comput. Sci.* Springer, 2014, pp. 229–245.
22. Tao, C., Diene, A., Tang, S., and Ding, J. Simple matrix scheme for encryption. In *PQCrypto 2013*, vol. 7932 of *Lecture Notes in Comput. Sci.* Springer, 2013, pp. 231–242.
23. Yang, B.-Y., and Chen, J.-M. All in the XL family: theory and practice. In *Information security and cryptology—ICISC 2004*, vol. 3506 of *Lecture Notes in Comput. Sci.* Springer, 2005, pp. 67–86.
24. Yasuda, T., and Sakurai, K. A multivariate encryption scheme with Rainbow. In *ICICS 2015*, vol. 9543 of *Lecture Notes in Comput. Sci.* Springer, 2016, pp. 222–236.

A Choice of the matrices B_1 and B_2

As mentioned in Section 3 of this paper, our technique allows the user of the scheme to choose the matrices B_1 and B_2 in an arbitrary way. However, in order to reduce the public key size and to speed up the encryption process of the scheme, we chose in this paper "cyclic" matrices B_1 and B_2.
In particular, we chose two random vectors $\mathbf{b}_1 \in \mathbb{F}^h$ and $\mathbf{b}_2 \in \mathbb{F}^{D-h}$. The first row of B_1 is given by $\mathbf{b}_1$, while the i-th row of B_1 corresponds to the cyclic right shift of $\mathbf{b}_1$ by $i-1$ positions ($i = 1, \ldots, m-d$). Similarly, the first row of B_2 is given by the vector B_2, while the i-th row of B_2 ($i = 2, \ldots, s$) corresponds to the cyclic right shift of $\mathbf{b}_2$ by $i-1$ positions. Algorithm 2 shows this generation process in algorithmic form.

Other possibilities to construct the matrices B_1 and B_2 include

- **Use of a Pseudo-Random-Number Generator (PRNG)**: When generating the matrices B_1 and B_2 using a PRNG with a small seed s of e.g. 128 bit, we have to store only s in order to recover B_1 and B_2. Therefore, the public key size of the scheme is reduced significantly. However, since there is no visible structure in the public key, we can not get a speed up in the encryption process.

Algorithm 2 Generation of the matrices B_1 and B_2 for CyclicSRP

```
Input: SRP parameters $q, d, o, r, l, s$
Output: matrices $B_1 \in \mathbb{F}^{(m-d)\times h}$ and $B_2 \in \mathbb{F}^{s\times(D-h)}$ for CyclicSRP
$m \leftarrow d + o + r + s$
$n \leftarrow d + o - l$
$D \leftarrow n \cdot (n + 1)/2$
$h \leftarrow d \cdot (d + 1)/2 + d \cdot (o - l)$
Choose randomly vectors $\mathbf{b}_1 \in \mathbb{F}^h$ and $\mathbf{b}_2 \in \mathbb{F}^{D-h}$
$B_1[1] \leftarrow \mathbf{b}_1$
for $i = 2$ to $o + r$ do
    $\mathbf{b}_1 \leftarrow \mathrm{CyclicRightShift}(\mathbf{b}_1, 1)$
    $B_1[i] \leftarrow \mathbf{b}_1$
end for
$\mathbf{b}_1 \leftarrow \mathbf{b}_1 \| \mathbf{b}_2$
for $i = 1$ to $s$ do
    $\mathbf{b}_1 \leftarrow \mathrm{CyclicRightShift}(\mathbf{b}_1, 1)$
    $B_1[o + r + i] \leftarrow (\mathbf{b}_1)_{1\ldots h}$
    $B_2[i] \leftarrow (\mathbf{b}_1)_{D-h+1\ldots D}$
end for
return $B_1$, $B_2$
```

– **Use of a Linear Feedback Shift Register (LFSR)**: When generating the matrices B_1 and B_2 using an LFSR, we have to store only the initial vector and the propagation polynomial of the LFSR in order to recover B_1 and B_2 and therefore observe a significant reduction in the public key size. Furthermore, we can use the structure of the public key to speed up the encryption process of the scheme. Additionally, linear recurring sequences offer good statistical properties which makes it hard to develop a structural attack against these schemes (see [19]).