Linear Recurring Sequences for the UOV Key Generation Revisited

Albrecht Petzoldt[1] and Stanislav Bulygin[2]

[1] Technische Universität Darmstadt, Department of Computer Science
Hochschulstraße 10, 64289 Darmstadt, Germany
apetzoldt@cdc.informatik.tu-darmstadt.de
[2] Center for Advanced Security Research Darmstadt - CASED
Mornewegstraße 32, 64293 Darmstadt, Germany
Stanislav.Bulygin@cased.de

Abstract. Multivariate cryptography is one of the main candidates to guarantee the security of communication in a post quantum era. While multivariate signature schemes are very fast and require only modest computational resources, the key sizes of such schemes are quite large. In [17] Petzoldt et al. proposed a way to use Linear Recurring Sequences (LRS's) for the key generation of the Unbalanced Oil and Vinegar (UOV) signature scheme by which they were able to reduce the public key size of this scheme by a factor of 7. In this paper we describe a modification of their scheme, which enables us not only to reduce the public key size, but also to speed up the verification process of the UOV scheme by a factor of 5.

Keywords: Multivariate Cryptography, UOV Signature Scheme, Key Size Reduction, Fast Verification.

1 Introduction

When quantum computers arrive, classical public-key cryptosystems like RSA and ECC will be broken [1]. The reason for this is Shor's algorithm [18] which solves number theoretic problems like integer factorization and discrete logarithms in polynomial time on a quantum computer. So, to guarantee the security of communication in a post quantum era, we need alternatives to those classical schemes. Besides lattice-, code-, and hash-based cryptosystems, multivariate cryptography seems to be a candidate for this.

Additionally to its (believed) resistance against quantum computer attacks, multivariate cryptosystems are very fast, especially for signatures [3, 5]. Furthermore they require only modest computational resources, which makes them attractive for the use on low-cost devices like smartcards and RFID chips. However, multivariate schemes are not widely used yet, mainly because of the large size of their public and private keys.

In [17] Petzoldt et al. proposed to use Linear Recurring Sequences (LRS) for the key generation of the Unbalanced Oil and Vinegar (UOV) Signature Scheme.

T. Kwon, M.-K. Lee, and D. Kwon (Eds.): ICISC 2012, LNCS 7839, pp. 441–455, 2013.

They did this by inserting a matrix B generated by an LRS into the coefficient matrix of the public key. Therefore, the Macauley matrix of the public key has the form $M_P = (B|C)$, where C is a matrix without visible structure. By doing so, they were able to decrease the public key size of UOV by a factor of 7, namely from 100 kB to about 14 kB.

In this paper we propose a variation of their scheme, which not only decreases the size of the public key, but also enables us to speed up the verification process. We show how to use the large structure of the matrix B to reduce the number of field multiplications needed during the verification process by a factor of 5. We derive our results both theoretically and show them using a C implementation of the scheme.

The structure of this paper is as follows: Section 2 gives a very short introduction on Linear Recurring Sequences (LRS). In Section 3 we give an overview on multivariate signature schemes and describe the UOV signature scheme. Section 4 reviews the approach of [16] to create UOV schemes with structured public keys . In Section 5 we describe our new approach in detail. Furthermore we look at the security of our scheme and consider the question how to choose the parameters of it. Section 6 demonstrates, how we can use the special structure of our polynomials to speed up the verification process. Finally, Section 7 presents the results of our computer experiments and Section 8 concludes the paper.

2 Linear Recurring Sequences (LRS)

In this section we repeat briefly results from the theory of linear recurring sequences (LRS's) needed in the following sections. For a more detailed introduction we refer to [13].

Definition 1. *Let L be a positive integer and $\gamma_1, \ldots, \gamma_L$ be elements of a finite field $\mathbb{F}$. A Linear Recurring Sequence (LRS) of length L is a sequence $\{s_1, s_2, \ldots\}$ of $\mathbb{F}$-elements satisfying the relation*

$$s_j = \gamma_1 \cdot s_{j-1} + \gamma_2 \cdot s_{j-2} + \cdots + \gamma_L \cdot s_{j-L} = \sum_{i=1}^{L} \gamma_i \cdot s_{j-i} \quad (\forall j > L). \tag{1}$$

The values $s_1, \ldots, s_L$ are called the initial values of the LRS.

Definition 2. *The connection polynomial of an LRS is defined as*

$$C(X) = \gamma_L \cdot X^L + \gamma_{L-1} \cdot X^{L-1} + \cdots + \gamma_1 \cdot X + 1 = \sum_{i=1}^{L} \gamma_i X^i + 1.$$

The LRS S is uniquely determined by its initial values $s_1, \ldots, s_L$ and the connection polynomial C (due to equation (1)). Therefore we denote the LRS by $S = LRS(s_1, \ldots, s_L, C)$.

3 Multivariate Public Key Cryptography

The basic idea behind multivariate cryptography is to choose a system $\mathcal{F}$ of m quadratic polynomials in n variables over a finite field $\mathbb{F}$ which can be easily inverted (central map). After that one chooses two affine invertible maps $\mathcal{S}$ and $\mathcal{T}$ to hide the structure of the central map. The public key of the cryptosystem is the composed quadratic map $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$ which is (hopefully) difficult to invert. The private key consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert $\mathcal{P}$.

Due to this construction, the security of multivariate cryptography is based on two mathematical problems:

Problem MQ: Solve the system $p_1 = \ldots = p_m = 0$, where each p_i is a quadratic polynomial in the n variables $x_1, \ldots, x_n$ with coefficients and variables in $\mathbb{F}$.

The MQ-problem is proven to be NP-hard even for quadratic polynomials over $GF(2)$ [10].

Problem EIP (Extended Isomorphism of Polynomials): Given a class of central maps $\mathcal{C}$ and a map $\mathcal{P}$ expressible as $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$, where $\mathcal{S}$ and $\mathcal{T}$ are affine maps and $\mathcal{F} \in \mathcal{C}$, find a decomposition of $\mathcal{P}$ of the form $\mathcal{P} = \mathcal{S}' \circ \mathcal{F}' \circ \mathcal{T}'$, with affine maps $\mathcal{S}'$ and $\mathcal{T}'$ and $\mathcal{F}' \in \mathcal{C}$.

In this paper we concentrate on the case of multivariate signature schemes. The standard process for signature generation and verification works as follows:

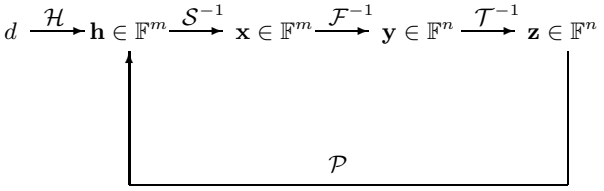

Fig. 1. Signature generation and verification

Signature Generation. To sign a document d, we use a hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{F}^m$ to compute the value $\mathbf{h} = \mathcal{H}(d) \in \mathbb{F}^m$. Then we compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{h})$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x})$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$. The signature of the document is $\mathbf{z} \in \mathbb{F}^n$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of the possibly many) pre-images of $\mathbf{x}$ under the central map $\mathcal{F}$.

Verification. To verify the authenticity of a document, one simply computes $\mathbf{h}' = \mathcal{P}(\mathbf{z})$ and the hash value $\mathbf{h} = \mathcal{H}(d)$ of the document. If $\mathbf{h}' = \mathbf{h}$ holds, the signature is accepted, otherwise rejected.

There are several ways how to build the central map $\mathcal{F}$ of multivariate schemes. In this paper we concentrate on the so called SingleField constructions. In contrast to BigField schemes like Matsumoto-Imai [14] and MiddleField schemes like ℓiC [8], here all the computations are done in one (relatively small) field. In the following subsection we describe one well known example for such a scheme in detail.

3.1 The UOV Signature Scheme

One way to create an easily invertible multivariate quadratic system is the principle of Oil and Vinegar, which was first proposed by J. Patarin in [15].

Let $\mathbb{F}$ be a finite field. Let o and v be two integers and set $n = o + v$. We set $V = \{1, \ldots, v\}$ and $O = \{v+1, \ldots, n\}$. We call $x_1, \ldots, x_v$ the Vinegar variables and $x_{v+1}, \ldots, x_n$ Oil variables and define o quadratic polynomials $f^{(k)}(\mathbf{x}) = f^{(k)}(x_1, \ldots, x_n)$ by

$$f^{(k)}(\mathbf{x}) = \sum_{i \in V,\ j \in O} \alpha_{ij}^{(k)} x_i x_j + \sum_{i,j \in V,\ i \leq j} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V \cup O} \gamma_i^{(k)} x_i + \eta^{(k)} \ (1 \leq k \leq o). \tag{2}$$

Note that Oil and Vinegar variables are not fully mixed, just like oil and vinegar in a salad dressing.

The map $\mathcal{F} = (f^{(1)}(\mathbf{x}), \ldots, f^{(o)}(\mathbf{x}))$ can be easily inverted. First, we choose the values of the v Vinegar variables $x_1, \ldots, x_v$ at random. Thus we get a system of o linear equations in the o variables $x_{v+1}, \ldots, x_n$ which can be solved e.g. by Gaussian Elimination. If the system does not have a solution, one has to choose other values of $x_1, \ldots, x_v$ and try again.

To hide the structure of $\mathcal{F}$ in the public key, one composes it with an affine map $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. Therefore, the public key has the form $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$. The private key consists of $\mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert the public key.

Remark: In opposite to other multivariate schemes the second affine map $\mathcal{S}$ is not needed for the security of UOV. So it can be dropped.

In his original paper [15] Patarin suggested to choose $o = v$ (Balanced Oil and Vinegar (OV)). After this scheme was broken by Kipnis and Shamir in [12], it was recommended in [11] to choose $v > o$ (Unbalanced Oil and Vinegar (UOV)).

The UOV signature scheme over $GF(2^8)$ is commonly believed to be secure for $o \geq 28$ equations [19] and $v = 2 \cdot o$ Vinegar variables. For UOV schemes over $GF(2^4)$ we need at least $o = 40$ equations and $v = 2 \cdot o$ Vinegar variables.

4 Improved versions of UOV

In this section we review the approach of [16] to create UOV-based schemes with a structured public key.

Recall that, in the case of the Unbalanced Oil and Vinegar signature scheme [11], the public key $\mathcal{P}$ is given as the composition of the central UOV-map $\mathcal{F}$ and an affine invertible map $\mathcal{T}$ (given by a matrix M_T and a vector c_T), i.e.

$$\mathcal{P} = \mathcal{F} \circ \mathcal{T}. \tag{3}$$

In [16] it is observed, that this equation (after fixing the affine map $\mathcal{T}$), leads to a linear relation between the coefficients of the quadratic monomials of $\mathcal{P}$ and $\mathcal{F}$ of the form

$$p_{ij}^{(k)} = \sum_{r=1}^{v} \sum_{s=r}^{n} \alpha_{ij}^{rs} \cdot f_{rs}^{(k)}, \tag{4}$$

where $p_{ij}^{(k)}$ and $f_{ij}^{(k)}$ are the coefficients of $x_i x_j$ in the k-th component of $\mathcal{P}$ and $\mathcal{F}$ respectively and the α_{ij}^{rs} are given as

$$\alpha_{ij}^{rs} = \begin{cases} t_{ri} \cdot t_{si} & (i = j) \\ t_{ri} \cdot t_{sj} + t_{rj} \cdot t_{si} & \text{otherwise} \end{cases}. \tag{5}$$

Here $t_{ij} \in \mathbb{F}$ denote the elements of the matrix M_T. Let $D := \frac{v \cdot (v+1)}{2} + ov$ be the number of non-zero quadratic terms in any component of $\mathcal{F}$ and $D' := \frac{n \cdot (n+1)}{2}$ be the number of quadratic terms in the public polynomials. Let M_P and M_F be the coefficient matrices of $\mathcal{P}$ and $\mathcal{F}$ respectively (w.r.t. the graded lexicographic ordering of monomials). The matrices M_P and M_F are divided into submatrices as shown in Figure 2. Note that, due to the absence of oil $\times$ oil terms in the central polynomials, we have a block of zeros in the middle of M_F.

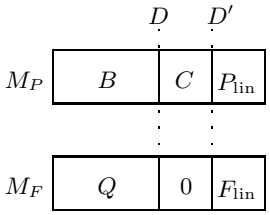

Fig. 2. Layout of the matrices M_P and M_F

Furthermore, the authors of [16] defined the so called transformation matrix $A_{UOV} \in \mathbb{F}^{D \times D}$ containing the coefficients α_{ij}^{rs} of equation (4)

$A_{UOV} = \left(\alpha_{ij}^{rs}\right)$ $(1 \leq r \leq v, r \leq s \leq n$ for the rows, $1 \leq i \leq v, i \leq j \leq n$ for the columns), i.e.

$$A_{UOV} = \begin{pmatrix} \alpha_{11}^{11} & \alpha_{12}^{11} & \dots & \alpha_{vn}^{11} \\ \alpha_{11}^{12} & \alpha_{12}^{12} & \dots & \alpha_{vn}^{12} \\ \vdots & & & \vdots \\ \alpha_{11}^{vn} & \alpha_{12}^{vn} & \dots & \alpha_{vn}^{vn} \end{pmatrix}. \tag{6}$$

With this notation, equation (4) yields

$$B = Q \cdot A_{UOV}. \tag{7}$$

If the matrix A_{UOV} is invertible, this equation has a solution for Q. Experiments indicate that this condition is fulfilled with overwhelming probability. We can then use Algorithm 1 to generate a key pair for UOV.

Algorithm 1. Alternative Key Generation for UOV schemes

Input: parameters $(\mathbb{F}, o, v)$
Output: UOV keypair $(\mathcal{F}, \mathcal{T}), \mathcal{P}$

1: $D \leftarrow \frac{v \cdot (v+1)}{2} + o \cdot v$
2: Choose an $o \times D$ matrix B (e.g. generated by an LRS).
3: Choose randomly an affine map $\mathcal{T}$ (represented by an $n \times n$-matrix M_T and an n-vector c_T). If M_T is not invertible, choose again.
4: Compute for $\mathcal{T}$ the corresponding transformation matrix A_{UOV} (using equations (5) and (6)). If A_{UOV} is not invertible, go back to step 2.
5: Solve the linear system given by equation (7) to get the matrix Q and therewith the quadratic coefficients of the central polynomials.
6: Choose the linear and constant terms of the central map $\mathcal{F}$ at random.
7: Compute the public key as $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$.
8: **return** $(\mathcal{F}, \mathcal{T}), \mathcal{P}$

5 Our Choice of B

The authors of [17] used a matrix B, whose elements were given by a single Linear Recurring Sequence, i.e. for a given LRS $S = (s_1, s_2, \dots)$ the matrix B was of the form

$$B^{(PB11)} = \begin{pmatrix} s_1 & s_2 & \dots & s_D \\ s_{D+1} & s_{D+2} & \dots & s_{2 \cdot D} \\ \vdots & & & \vdots \\ s_{(o-1) \cdot D+1} & s_{(o-1) \cdot D+2} & \dots & s_{o \cdot D} \end{pmatrix} \tag{8}$$

To guarantee the security of the scheme, they had to choose a Linear Recurring Sequence of length $L \geq o$.

In our new scheme, we use not only one, but o different Linear Recurring Sequences. The goal of this strategy is to reduce the lengths of the single LRS's, which will later help us to speed up the verification process of the scheme (see Section 6). In fact, we will use Linear Recurring Sequences of length 1.

We choose randomly two vectors $\alpha,\ \gamma \in \mathbb{F}^o$ and define for each $i = 1, \ldots, o$ a univariate polynomial C_i by $C_i(X) = \gamma_i \cdot X + 1$. For $i = 1, \ldots, o$ we compute the first D elements of the Linear Recurring Sequence $S^{(i)} = (s_1^{(i)}, s_2^{(i)}, \ldots) = LRS(\alpha_i, C_i)$ and put this sequence into the i-th row of the matrix B. Therefore, the matrix B will have the following structure:

$$B = \begin{pmatrix} s_1^{(1)} & s_2^{(1)} & \ldots & s_D^{(1)} \\ s_1^{(2)} & s_2^{(2)} & \ldots & s_D^{(2)} \\ \vdots & & & \vdots \\ s_1^{(o)} & s_2^{(o)} & \ldots & s_D^{(o)} \end{pmatrix}. \tag{9}$$

We denote the scheme obtained by using this matrix B and Algorithm 1 by UOVLRS2.

5.1 Choice of α and γ

First, we look at the question what happens if two elements of the vector γ, say γ_i and γ_j $(i \neq j)$ are equal.

Theorem 1. *If $\gamma_i = \gamma_j$ for $i \neq j \in \{1, \ldots, o\}$, the homogeneous quadratic parts of the polynomials $p^{(i)}$ and $p^{(j)}$ are linearly dependent.*

Proof. If $\gamma_i = \gamma_j$ for $i \neq j \in \{1, \ldots, o\}$, the two rows $B[i]$ and $B[j]$ are linearly dependent. Since we have $Q = B \cdot A^{-1}$ (c.f. equation (7)), the same holds for $Q[i]$ and $Q[j]$ (see Figure 2). Note that this matrix contains all the private coefficients of quadratic terms, which means that the homogeneous quadratic parts of the i-th and j-th central polynomials are linearly dependent. Since during the key generation of UOV the rows of the central map $\mathcal{F}$ are not mixed, the same holds for the homogeneous quadratic part of the i-th and j-th public polynomial. □

Theorem 1 states that by computing $p^{(i)} - \frac{\alpha_i}{\alpha_j} \cdot p^{(j)}$ the attacker will get a linear equation in the system variables, which means that he can reduce the number of variables in the quadratic system by 1. We can conclude

Corollary 1. *Attacking an instance of UOVLRS2 with m equations and $t < m$ different values in the vector γ is only as hard as solving a (UOVLRS2) system of t equations.*

To check this theoretical result, we created instances of UOVLRS2 for different values of o and v and different types of vectors γ and solved the resulting public systems with MAGMA v.2-13.10 (with fixing of v variables to create determined systems).

Table 1. Running time of direct attacks with MAGMA

t [1]	(o, v)	(9,18)	(10,20)	(11,22)	(12,24)	(13,26)	(14,28)
9	time (s)	5.4	5.7	5.7	5.8	5.8	5.9
10	time (s)	———	38.9	40.8	41.8	43.0	44.6
11	time (s)	———	———	287.3	301.4	309.8	315.2

[1] number of different values in γ

To achieve the optimal security level, the elements of the vector γ must be pairwise distinct. Furthermore, all the elements have to be $\neq 0$.

Remark: The above condition gives a lower bound to the cardinality of the underlying field. In particular, we can not define our scheme over $GF(2^4)$.

On the contrary, there seem to be no major conditions for the choice of the vector α. We have to ensure only that $\alpha_i \in \mathbb{F} \setminus \{0\} \;\; \forall i = 1, \ldots, o$. For simplicity we choose $\alpha = (1, \ldots, 1)$.[2] Therefore, we get a matrix B of the Vandermonde-type:

$$B = \begin{pmatrix} 1 & \gamma_1 & \gamma_1^2 & \ldots & \gamma_1^{D-1} \\ 1 & \gamma_2 & \gamma_2^2 & \ldots & \gamma_2^{D-1} \\ \vdots & & & & \vdots \\ 1 & \gamma_o & \gamma_o^2 & \ldots & \gamma_o^{D-1} \end{pmatrix} \tag{10}$$

which can be used in Algortihm 1 to generate a key pair of UOVLRS2.

5.2 Security

As mentioned above, the matrix B of our scheme is of the Vandermonde type. If the elements of the vector γ are pairwise distinct, there is not any relationship between the rows of B at all. This is in contrast to the schemes of [16] and [17] and prevents therefore possible attacks against schemes of this type which use such relationships. Furthermore this is very similar to the case of standard UOV, which seems to show that direct attacks against our scheme are as difficult as direct attacks against standard UOV. Further evidence for this result was given by experiments with MAGMA [2].

Furthermore we checked experimentally the security of our scheme against other attacks affecting UOV-like schemes, including

- UOV attack of Kipnis and Shamir [11]
- UOV Reconciliation attack [7]

and found that these attacks cannot use the structure in our systems. Details on these experiments can be found in the appendix of this paper.

[2] In fact, the attacker is allowed to multiply each public polynomial $p^{(i)}$ $(i = 1, \ldots, o)$ by a number $a_i \in \mathbb{F} \setminus \{0\}$ of his choice. By doing so, he can produce a vector α' of this form.

6 The Verification Process

The central part of the verification process for multivariate signature schemes is the evaluation of the public polynomials. Normally this is done as follows: For a given (valid or invalid) signature $\mathbf{z} = (z_1, \ldots, z_n) \in \mathbb{F}^n$ one first computes an $\frac{(n+1)\cdot(n+2)}{2}$ vector mon, which contains the values of all monomials of degree ≤ 2, i.e.

$$\text{mon} = (z_1^2, z_1 z_2, \ldots, z_n^2, z_1, \ldots, z_n, 1). \tag{11}$$

Then we have

$$\mathcal{P}(\mathbf{z}) = \begin{pmatrix} M_P[1] \cdot \text{mon}^T \\ \vdots \\ M_P[o] \cdot \text{mon}^T \end{pmatrix}, \tag{12}$$

with $M_P[i]$ being the i-th row of the Macauley matrix M_P.

For our new scheme, the following strategy seems to be more promising:

6.1 Notations

Let $\mathbf{h} = (h_1, \ldots, h_o)$ be the hash value of the signed message.
The public polynomials can be written as

$$p^{(k)}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i}^{n} p_{ij}^{(k)} \cdot x_i x_j + \sum_{i=1}^{n} p_i^{(k)} \cdot x_i + p_0^{(k)} \quad (k = 1, \ldots, o). \tag{13}$$

For $k = 1, \ldots, o$ we define upper triangular matrices $MP^{(k)}$ by

$$MP^{(k)} = \begin{pmatrix} p_{11}^{(k)} & p_{12}^{(k)} & p_{13}^{(k)} & \cdots & p_{1n}^{(k)} & p_1^{(k)} \\ 0 & p_{22}^{(k)} & p_{23}^{(k)} & \cdots & p_{2n}^{(k)} & p_2^{(k)} \\ 0 & 0 & p_{33}^{(k)} & & p_{3n}^{(k)} & p_3^{(k)} \\ \vdots & & \ddots & & \vdots & \vdots \\ 0 & 0 & \ldots & 0 & p_{nn}^{(k)} & p_n^{(k)} \\ 0 & 0 & \ldots & 0 & 0 & p_0^{(k)} \end{pmatrix}. \tag{14}$$

For a (valid or invalid) signature $\mathbf{z} = (z_1, \ldots, z_n)$ of the message we define the extended signature vector

$$\text{sign} = (z_1, \ldots, z_n, 1). \tag{15}$$

With this notation we can write the verification process in the following form

$$\text{accept the signature } \mathbf{z} \iff \text{sign} \cdot MP^{(k)} \cdot \text{sign}^T = h_k \ \forall k \in \{1, \ldots, o\}. \tag{16}$$

In the following subsection we consider the question how we can evaluate this equation more efficiently for our scheme.

6.2 Verification of UOVLRS2

In the case of UOVLRS2, the matrices $MP^{(k)}$ are of the form shown in Figure 3.

$$MP^{(k)} = \begin{pmatrix}
\boxed{1} & \boxed{\gamma_k} & \boxed{\gamma_k^2} & \cdots & \boxed{\gamma_k^{v-2}} & \boxed{\gamma_k^{v-1}} & \boxed{\gamma_k^{v}} & \cdots & \boxed{\gamma_k^{n-2}} & \boxed{\gamma_k^{n-1}} & \star \\
0 & \boxed{\gamma_k^n} & \boxed{\gamma_k^{n+1}} & \cdots & \boxed{\gamma_k^{n+v-3}} & \boxed{\gamma_k^{n+v-2}} & \boxed{\gamma_k^{n+v-1}} & \cdots & \boxed{\gamma_k^{2n-3}} & \boxed{\gamma_k^{2n-2}} & \star \\
0 & 0 & \boxed{\gamma_k^{2n-1}} & \cdots & \boxed{\gamma_k^{2n+v-5}} & \boxed{\gamma_k^{2n+v-4}} & \boxed{\gamma_k^{2n+v-3}} & \cdots & \boxed{\gamma_k^{3n-5}} & \boxed{\gamma_k^{3n-4}} & \star \\
\vdots & & \ddots & & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\
0 & & \cdots & 0 & \boxed{\gamma_k^{D-2o-2}} & \boxed{\gamma_k^{D-2o-1}} & \boxed{\gamma_k^{D-2o}} & \cdots & \boxed{\gamma_k^{D-o-4}} & \boxed{\gamma_k^{D-o-3}} & \star \\
0 & & \cdots & & 0 & \boxed{\gamma_k^{D-o-1}} & \boxed{\gamma_k^{D-o}} & \cdots & \boxed{\gamma_k^{D-2}} & \boxed{\gamma_k^{D-1}} & \star \\
0 & & \cdots & & & 0 & \star & \cdots & \star & \star & \star \\
\vdots & & & & & & \ddots & & \vdots & \vdots & \vdots \\
 & & & & & & & \ddots & \vdots & \vdots & \vdots \\
0 & & \cdots & & \cdots & & \cdots & & 0 & \star & \star \\
0 & & \cdots & & \cdots & & \cdots & & & 0 & \star
\end{pmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ \\ \\ v \\ \\ \\ \\ \\ \end{matrix}$$

Fig. 3. Matrices $MP^{(k)}$ for UOVLRS2

We have

$$MP_{ij}^{(k)} = \gamma_k \cdot MP_{i,j-1}^{(k)} \ \forall i \in \{1, \ldots, v\},\ j \in \{i+1, \ldots, n\},\ k \in \{1, \ldots, o\}. \quad (17)$$

Therefore we get

$$(\mathrm{sign}_1, \ldots, \mathrm{sign}_i) \cdot \begin{pmatrix} MP_{1,j}^{(k)} \\ MP_{2,j}^{(k)} \\ \vdots \\ MP_{i,j}^{(k)} \end{pmatrix} = \gamma_k \cdot (\mathrm{sign}_1, \ldots, \mathrm{sign}_i) \cdot \begin{pmatrix} MP_{1,j-1}^{(k)} \\ MP_{2,j-1}^{(k)} \\ \vdots \\ MP_{i,j-1}^{(k)} \end{pmatrix} \quad (18)$$

$$\forall i \in \{1, \ldots v\}, \quad j \in \{i+1, \ldots, n\},\ k \in \{1, \ldots, o\}.$$

The boxes in Figure 3 illustrate this equation: Boxes with continuous lines show the vector $(MP_{1,j-1}^{(k)}, \ldots, MP_{i,j-1}^{k})^T$ on the right hand side of equation (18), while the boxes with dotted lines show the vector $(MP_{1,j}^{(k)}, \ldots, MP_{i,j}^{(k)})^T$ on the left hand side. Any box with dotted lines can be computed by multiplying the corresponding box with continuous lines by γ_k.

We can use this fact to speed up the verification process of UOVLRS2 by a large factor (see Algorithm 2).

Algorithm 2 works as follows:

From line 2 to 14 the public polynomials are evaluated. From line 3 to 12 we hereby compute the matrix vector product $\mathrm{sign} \cdot MP^{(k)} \cdot$ whose result is stored

Algorithm 2. Verification process for UOVLRS2

Input: signature $\mathbf{z} \in \mathbb{F}^n$, hash value $\mathbf{h} \in \mathbb{F}^m$
Output: Boolean value TRUE or FALSE

1: $\text{sign} \leftarrow (z_1, \ldots, z_n, 1)$
2: **for** $k = 1$ to o **do**
3: $\quad \text{temp}_1 \leftarrow \text{sign}_1$
4: $\quad$ **for** $j = 2$ to v **do**
5: $\quad\quad \text{temp}_j \leftarrow \gamma_k \cdot \text{temp}_{j-1} + MP_{jj}^{(k)} \cdot \text{sign}_j$
6: $\quad$ **end for**
7: $\quad a \leftarrow \text{temp}_v$
8: $\quad$ **for** $j = v + 1$ to n **do**
9: $\quad\quad a \leftarrow \gamma_k \cdot a$
10: $\quad\quad \text{temp}_j \leftarrow a + \sum_{i=v+1}^{j} MP_{ij}^{(k)} \cdot \text{sign}_i$
11: $\quad$ **end for**
12: $\quad \text{temp}_{n+1} \leftarrow \sum_{i=1}^{n+1} MP_{i,n+1}^{(k)} \cdot \text{sign}_\text{i}$
13: $\quad h'_k \leftarrow \sum_{i=1}^{n+1} \text{temp}_i \cdot \text{sign}_i$
14: **end for**
15: **if** $h_k = h'_k \ \forall k \in \{1, \ldots, o\}$ **then**
16: $\quad$ **return** TRUE
17: **else**
18: $\quad$ **return** FALSE
19: **end if**

in the vector temp. In line 5 and line 9-10 we hereby use the special structure of our public key, which allows us to compute each temp_i $(i = 2, \ldots, n)$ using only two multiplications. Finally, in line 13 of the algorithm, we compute the scalar product of temp and sign.

In line 15 to 19 we test, if the result is equal to the hash value of the message.

Computational Effort. To evaluate $\mathcal{P}$ in the standard way (i.e. by using equations (11) and (12)), one needs

$$\frac{n+1}{2} \cdot (n + o \cdot (n + 2)) \text{ field multiplications.} \tag{19}$$

Algorithm 2 needs (for each iteration of the main loop)

- in the first loop (step 4 to 6) $2 \cdot (v - 1)$ field multiplications,
- in the second loop (step 8 to 11) $o + \frac{o \cdot (o+1)}{2}$ field multiplications,
- in step 12 $n + 1$ field multiplications,
- and in step 13 again $n + 1$ field multiplications.

Therefore, to evaluate equation (16) (o iterations of the main loop), Algorithm 2 needs

$$o \cdot \left(3 \cdot n + v + \frac{o \cdot (o+1)}{2}\right) \text{ field multiplications.} \tag{20}$$

For $\mathbb{F} = GF(2^8)$, $(o, v) = (28, 56)$ this means a reduction of the number of field multiplications needed during the verification process by a factor of 5.3.

7 Parameters and Experiments

Based on our security analysis (see Subsection 5.2 and Appendix A), we propose for our scheme the same parameters as for the standard UOV scheme, namely

$$\mathbb{F} = GF(256),\ (o, v) = (28, 56).$$

The elements of the vector $\gamma \in \mathbb{F}^o$ are chosen pairwise distinct and we set $\alpha = (1, \ldots, 1) \in \mathbb{F}^o$.

To check our theoretical results regarding the verification process, we created a straightforward C implementation of our scheme and the standard UOV. Table 2 shows the results:

Table 2. Comparison of our scheme with standard UOV

Scheme	private key size (kB)	hash length (bit)	signature length (bit)	public key size (kB)	public key red. factor	verification time ms	verification time red. factor
UOV$(2^8, 28, 56)$	96.6	224	672	99.9	-	0.99	-
UOVLRS2$(2^8, 28, 56)$	96.6	224	672	13.5	7.4	0.18	5.5
UOV$(2^8, 30, 60)$	117.0	240	720	122.6	-	1.21	-
UOVLRS2$(2^8, 30, 60)$	117.0	240	720	16.4	7.5	0.21	5.7

8 Conclusion and Future Work

In this paper we proposed a variation of the UOVLRS scheme of [17], which not only achieves a similar reduction of the public key size but also speeds up the verification process of UOV by a large factor. In particular, we achieved a reduction of the public key size of UOV by a factor of 7.5 and a speed up factor of 5.5 for the verification process. We showed the latter both theoretically and by a C implementation of the schemes. Furthermore, experiments seem to show that the security of UOV is not weakened by our modifications.

Future work includes the extension of our ideas to the Rainbow Signature scheme [6] and the implementation of the scheme on hardware. Furthermore we want to apply our techniques to the QUAD stream cipher [4] to speed up its key stream generation process.

Acknowledgements. We thank the anonymous reviewers for their comments which helped to improve the paper. The first author thanks the Horst-Görtz Foundation for financial support, while the second author is funded by DFG grant BU630/22-1.

References

[1] Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.): Post-Quantum Cryptography. Springer, Heidelberg (2009)

[2] Bosma, W., Cannon, J., Playoust, C.: The Magma algebra system. I. The user language. J. Symbolic Comput. 24(3-4), 235–265 (1997)

[3] Bogdanov, A., Eisenbarth, T., Rupp, A., Wolf, C.: Time-area optimized public-key engines: -cryptosystems as replacement for elliptic curves? In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 45–61. Springer, Heidelberg (2008)

[4] Berbain, C., Gilbert, H., Patarin, J.: QUAD: A Practical Stream Cipher with Provable Security. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 109–128. Springer, Heidelberg (2006)

[5] Chen, A.I.-T., Chen, M.-S., Chen, T.-R., Cheng, C.-M., Ding, J., Kuo, E.L.-H., Lee, F.Y.-S., Yang, B.-Y.: SSE implementation of multivariate pkcs on modern x86 cpus. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 33–48. Springer, Heidelberg (2009)

[6] Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005)

[7] Ding, J., Yang, B.-Y., Chen, C.-H.O., Chen, M.-S., Cheng, C.-M.: New Differential-Algebraic Attacks and Reparametrization of Rainbow. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 242–257. Springer, Heidelberg (2008)

[8] Ding, J., Wolf, C., Yang, B.-Y.: ℓ-invertible Cycles for Multivariate Quadratic Public Key Cryptography. In: Okamoto, T., Wang, X. (eds.) PKC 2007. LNCS, vol. 4450, pp. 266–281. Springer, Heidelberg (2007)

[9] Faugère, J.C.: A new efficient algorithm for computing Groebner bases (F4). Journal of Pure and Applied Algebra 139, 61–88 (1999)

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

[11] Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar Signature Schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999)

[12] Kipnis, A., Shamir, A.: Cryptanalysis of the Oil and Vinegar Signature scheme. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 257–266. Springer, Heidelberg (1998)

[13] Lidl, R., Niederreiter, H.: Introduction to Finite Fields and their Applications. Cambridge University Press (1986)

[14] Matsumoto, T., Imai, H.: Public Quadratic Polynomial-Tuples for efficient Signature-Verification and Message-Encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)

[15] Patarin, J.: The oil and vinegar signature scheme. Presented at the Dagstuhl Workshop on Cryptography (September 1997)

[16] Petzoldt, A., Bulygin, S., Buchmann, J.: A Multivariate Signature Scheme with a partially cyclic public key. In: Proceedings of SCC 2010, pp. 229–235 (2010)

[17] Petzoldt, A., Bulygin, S., Buchmann, J.: Linear Recurring Sequences for the UOV Key Generation. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 335–350. Springer, Heidelberg (2011)

[18] Shor, P.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. 26(5), 1484–1509
[19] Thomae, E., Wolf, C.: Solving underdetermined Systems of Multivariate Quadratic Equations Revisited. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 156–171. Springer, Heidelberg (2012)

A Details of the Experiments

In this section we present the results of our experiments with known attacks against the UOV scheme. In particular, we test the security of our scheme against

- Direct attacks
- UOV-Reconciliation attack
- UOV-attack

A.1 Direct Attacks

In a direct attack an attacker tries to solve the public system $\mathcal{P}(x) = h$ by a system solver like XL or a Gröbner Basis method. Direct attacks can be used against each multivariate scheme as a message recovery attack (encryption schemes) or a signature forgery attack (signature schemes). To check the security of the UOVLRS2 scheme under direct attacks, we carried out a number of experiments with MAGMA [2] v.2-13.10, which contains an efficient implementation of Faugère's F_4 algorithm [9] to compute Gröbner Bases. For each of the parameter sets listed in Table 3 we created 100 instances of UOV and UOVLRS2 and solved the public systems using the MAGMA command `Variety`.

Table 3. Running time of the direct attack against UOV and UOVLRS2

parameters $(2^8, o, v)$	(9,18)	(10,20)	(11,22)	(12,24)	(13,26)	(14,28)
UOV	5.5 s	40.0 s	289.2 s	2,383 s	18,928 s	196,638 s
UOVLRS2	5.4 s	39.9 s	288.6 s	2,378 s	18,917 s	195,963 s

A.2 UOV-Reconciliation Attack

In the UOV-Reconciliation attack [7] the attacker tries to find an affine transformation which brings the public key in the form of a UOV central map (i.e. no Oil $\times$ Oil terms). To do this, the attacker has to solve a number of multivariate quadratic systems. The complexity of the attack is mainly given by the complexity of solving the first of these systems, which contains o equations in v variables. Table 4 shows the time, MAGMA needs for solving this first system for standard UOV and UOVLRS2.

Table 4. Running time of the UOV-Reconciliation attack against UOV and UOVLRS2

parameters $(2^8, o, v)$	(9,18)	(10,20)	(11,22)	(12,24)	(13,26)	(14,28)
UOV	5.5 s	40.1 s	289.3 s	2,381 s	18,924 s	196,712 s
UOVLRS2	5.4 s	39.9 s	286.8 s	2,379 s	18,923 s	196,683 s

A.3 UOV Attack

In the UOV attack of Kipnis and Shamir [12] the attacker tries to reconstruct the essential parts of the affine transformation $\mathcal{T}$ (i.e. the parts which mix Oil and Vinegar variables). To do this, he tries to find the space $\mathcal{T}^{-1}(\mathcal{O})$, where $\mathcal{O}$ is the so called Oil space

$$\mathcal{O} = \{x \in \mathbb{F}^n : x_1 = \ldots = x_v = 0\}.$$

This can be done by looking at the invariant subspaces of the linear maps $W = P_i^{-1} \cdot \sum_{j=1}^{o} P_j$, where P_j is the symmetric matrix associated with the homogeneous part of the j-th public polynomial. Table 5 shows the base 2-logarithm of the number of matrices W we had to test until finding a basis of $\mathcal{T}^{-1}(\mathcal{O})$.

Table 5. Results of the experiments with the UOV attack

parameters $(2^8, o, v)$	(2,4)	(3,6)	(4,8)	(5,10)
UOV	16.1	24.3	32.2	40.0
UOVLRS2	16.0	24.1	32.0	39.9

As the Tables 3 - 5 show, known attacks against the UOV signature scheme cannot use the special structure of our public keys. Of course, this is no proof that no dedicated attacks against our scheme exist. However, as long as no such attack is known, we believe our scheme to be secure and propose for it the same parameters as for the standard UOV scheme (see Section 7).