Practical key-recovery attack on MQ-Sign

Thomas Aulbach[1], Simona Samardjiska[2], and Monika Trimoska[2]

[1] University of Regensburg, Regensburg, Germany
[2] Radboud Universiteit, Nijmegen, The Netherlands
thomas.aulbach@ur.de,{simonas,mtrimoska}@cs.ru.nl

Abstract. This note describes attacks on the UOV-based signature scheme called MQ-Sign. In more detail, it presents a polynomial-time key-recovery attack on the variants MQ-Sign-RS and MQ-Sign-SS and an improved direct attack on the variant MQ-Sign-SR. The scheme is a first-round candidate in the Korean Post-Quantum Cryptography Competition. Our attack exploits the sparsity of the secret central polynomials in combination with the specific structure of the secret linear map S. We provide a verification script for the polynomial-time attack, that recovers the secret key in less than seven seconds for security level V. Furthermore, we added an implementation of the non-guessing part of the direct attack, confirming our complexity estimates.

1 Introduction

The lack of diversity of hardness assumptions motivated NIST's announcement of reopening the call for post-quantum digital signature proposals, specifying the need for shorter signatures with fast verification. Multivariate cryptography is a contender in this ongoing search for post-quantum digital signature schemes. Since many schemes in multivariate cryptography make use only of quadratic polynomials, public key cryptosystems in the multivariate family are often referred to as $\mathcal{MQ}$ public key cryptosystems. However, the security of some of the $\mathcal{MQ}$ systems does not rely directly on the hardness of the Multivariate Quadratic polynomial ($\mathcal{MQ}$) problem, but rather on the (non)possibility of exploiting a planted trapdoor. The trapdoor usually consists of the knowledge of a so-called central map $\mathcal{F}$ that is easy to invert and a linear or affine transformation $\mathcal{S}$.

One of the most studied trapdoor-based systems is the Unbalanced Oil and Vinegar (UOV) signature scheme [8]. The UOV construction results in short signatures and signing time, but large public and secret keys. Several recent efforts are focused on developing a UOV-based signature scheme with additional structure that results in smaller keys, but without compromising the security. One of those efforts is the MQ-Sign [13] signature scheme submitted to the Korean Post-Quantum Cryptography Competition[3]. The main idea behind MQ-Sign is to have a sparse central map in order to reduce the secret key. In this

[3] www.kpqc.or.kr

work, we show how the property of using sparse polynomials can be exploited to develop a polynomial time key-recovery attack on the variants MQ-Sign-RS and MQ-SIGN-SS. Our attack focuses on recovering the linear transformation $\mathcal{S}$, which allows to subsequently compute the central map $\mathcal{F}$. Additionally, we introduce an improved direct attack with still exponential running time on the variant MQ-Sign-SR, that exploits a bilinear substructure in the sparse secret polynomials.

2 Preliminaries

Throughout the text, $\mathbb{F}_q$ will denote the finite field of q elements, and $\mathrm{GL}_n(\mathbb{F}_q)$ and $\mathrm{AGL}_n(\mathbb{F}_q)$ will denote respectively the general linear group and the general affine group of degree n over $\mathbb{F}_q$. We will also use the notation $\mathbf{x} = (x_1, \ldots, x_n)^\intercal$ for the vector $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$. Similarly, the entries of a matrix A are denoted by $A_{[ij]}$.

2.1 Multivariate signatures

First, we recall the general principle of $\mathcal{MQ}$ public key cryptosystems.

A typical $\mathcal{MQ}$ public key cryptosystem relies on the knowledge of a trapdoor for a particular system of polynomials over the field $\mathbb{F}_q$. The public key of the cryptosystem is usually given by a multivariate quadratic map $\mathcal{P} = (\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(m)}) : \mathbb{F}_q^n \to \mathbb{F}_q^m$, where

$$\mathcal{P}^{(k)}(x_1, \ldots, x_n) = \sum_{1 \le i \le j \le n} \gamma_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(k)} x_i + \alpha^{(k)}$$

for some coefficients $\gamma_{ij}^{(k)}, \beta_i^{(k)}, \alpha^{(k)} \in \mathbb{F}_q$. It is obtained by obfuscating a structured central map

$$\mathcal{F} : (x_1, \ldots, x_n) \in \mathbb{F}_q^n \to \left(\mathcal{F}^{(1)}(x_1, \ldots, x_n), \ldots, \mathcal{F}^{(m)}(x_1, \ldots, x_n)\right) \in \mathbb{F}_q^m,$$

using two bijective affine mappings $\mathcal{S}, \mathcal{T} \in \mathrm{AGL}_n(q)(\mathbb{F}_q)$ that serve as a sort of mask to hide the structure of $\mathcal{F}$. The public key is defined as

$$\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{S}.$$

The mappings $\mathcal{S}$ and $\mathcal{T}$ are part of the private key s. Besides them, the private key may also contain other secret parameters that allow creation, but also easy inversion of the transformation $\mathcal{F}$. Without loss of generality, we can assume that the private key is $s = (\mathcal{F}, \mathcal{S}, \mathcal{T})$.

Signature Generation. To generate a signature for a message d, the signer uses a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ and computes recursively $\mathbf{x} = \mathcal{T}^{-1}(\mathbf{w}) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$, and $\mathbf{z} = \mathcal{S}^{-1}(\mathbf{y})$. The signature of the message d is $\mathbf{z} \in \mathbb{F}^n$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of possibly many) preimages 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 a message d, one computes $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^m$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise it is rejected.

The standard signature generation and verification process of a multivariate signature scheme works as shown in Figure 1.

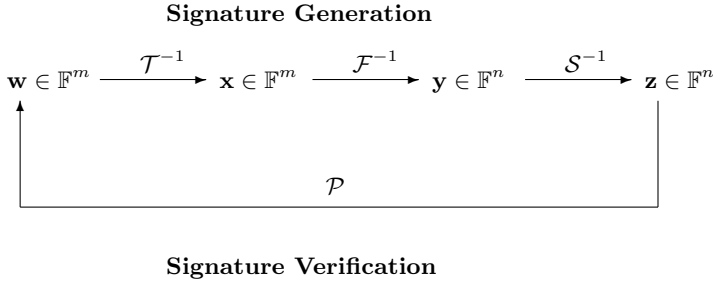

Fig. 1. General workflow of multivariate signature schemes.

2.2 Unbalanced Oil and Vinegar

The Unbalanced Oil and Vinegar signature scheme is one of the oldest multivariate signature schemes. It was proposed by Kipnis and Patarin at EUROCRYPT'99 [8] as a modification of the oil and vinegar scheme of Patarin [11] that was broken by Kipnis and Shamir in 1998 [9].

The characteristic of the oil and vinegar construction is in the special structure of the central map in which the variables are divided in two distinct sets, vinegar variables and oil variables. The vinegar variables are combined quadratically with all of the variables, while the oil variables are only combined quadratically with vinegar variables and not with other oil variables. Formally, the central map is defined as $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, with central polynomials

$$\mathcal{F}^{(k)}(x_1, \ldots, x_n) = \sum_{i \in V, j \in V} \gamma_{ij}^{(k)} x_i x_j + \sum_{i \in V, j \in O} \gamma_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(k)} x_i + \alpha^{(k)} \quad (1)$$

where $n = v + m$, and $V = \{1, \ldots, v\}$ and $O = \{v + 1, \ldots, n\}$ denote the index sets of the vinegar and oil variables, respectively.

It can be shown that if an oil an vinegar central map is used in the standard $\mathcal{MQ}$ construction the affine mapping $\mathcal{T}$ does not add to the security of the scheme and is therefore not necessary. Hence the secret key consists of a linear transformation $\mathcal{S}$ and central map $\mathcal{F}$, while the public key is defined as $\mathcal{P} = \mathcal{F} \circ \mathcal{S}$. In order to sign a message, we need to find a preimage of $\mathcal{F}$. This can be done

by simply fixing the vinegar variables to some random values. In this way, we obtain a system of m linear equations in m variables, which has a solution with probability around $1 - 1/q$. If the obtained system does not have a solution, we repeat the procedure with different values for the vinegar variables.

Key Generation. It was shown in [12] that for any instance of a UOV secret key $(\mathcal{F}, \mathcal{S})$, there exists an equivalent secret key $(S, \mathcal{F})$ with

$$S = \begin{pmatrix} I_{v\times v} & S_1 \\ 0_{m\times v} & I_{m\times m} \end{pmatrix}. \tag{2}$$

Furthermore, the quadratic polynomials of the central map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ can be represented using upper triangular matrices $F^{(1)}, \ldots, F^{(m)} \in \mathbb{F}_q^{n\times n}$ where each nonzero coefficient (i, j) in $F^{(k)}$ corresponds to the nonzero coefficient of $x_i x_j$ in $\mathcal{F}^{(k)}$. Note that the $m \times m$ block on the bottom right of these matrices is empty, since the polynomials of the central map have no quadratic oil terms. Thus, these matrices contain an upper triangular block $F_1^{(k)} \in \mathbb{F}_q^{v\times v}$ and a block $F_2^{(k)} \in \mathbb{F}_q^{v\times m}$ on the top right. In other words, the matrices are of the form:

$$F^{(k)} = \begin{pmatrix} F_1^{(k)} & F_2^{(k)} \\ 0 & 0 \end{pmatrix}.$$

Thus, in order to obtain a key pair, it suffices to first randomly generate $(S_1, F^{(1)}, \ldots, F^{(m)})$ and then compute $(P^{(1)}, \ldots, P^{(m)})$ by evaluating $P^{(k)} = S^\top F^{(k)} S$ and bringing the resulting matrices to upper triangular form.

2.3 MQ-Sign

MQ-Sign is a signature scheme based on UOV. The scheme uses inhomogenous polynomials and each polynomial of the central map can be written as

$$\mathcal{F}^{(k)} = \mathcal{F}_V^{(k)} + \mathcal{F}_{OV}^{(k)} + \mathcal{F}_{L,C}^{(k)}$$

where $\mathcal{F}_V^{(k)} = \sum_{i\in V, j\in V} \gamma_{ij}^{(k)} x_i x_j$ and $\mathcal{F}_{OV}^{(k)} = \sum_{i\in V, j\in O} \gamma_{ij}^{(k)} x_i x_j$. These can alternatively be referred to as the vinegar-vinegar quadratic part and the vinegar-oil quadratic part. Finally, $\mathcal{F}_{L,C}^{(k)}$ refers to the linear and constant part of the polynomials. In the following, we ignore the linear and constant parts, since our attack does not use them. The goal of MQ-Sign is to reduce the size of the secret key compared to traditional UOV. This is achieved by using sparse polynomials for the quadratic part of the central map. The quadratic homogenous part of the sparse polynomials is defined as $\mathcal{F}_V^{(k)} + \mathcal{F}_{OV}^{(k)}$ such that

$$\begin{aligned} \mathcal{F}_V^{(k)} &= \sum_{i=1}^{v} \alpha_i^k x_i x_{(i+k-1(\bmod v))+1} \\ \mathcal{F}_{OV}^{(k)} &= \sum_{i=1}^{v} \beta_i^k x_i x_{(i+k-2(\bmod m))+v+1}. \end{aligned} \tag{3}$$

The size of the secret key is thus reduced to $2vm$ field elements.

The MQ-Sign proposal provides a parameter selection for four variations of the scheme: MQ-Sign-SS, MQ-Sign-RS, MQ-Sign-SR and MQ-Sign-RR. The first S/R in the suffix specifies whether $\mathcal{F}_V$ is defined with sparse or random polynomials. The second S/R refers to the same property, but for $\mathcal{F}_{OV}$. Note that the variation MQ-Sign-RR corresponds to the standard UOV scheme defined with inhomogenous polynomials.

3 An efficient key-recovery on sparse $\mathcal{F}_{OV}$

In the following, we consider $\mathcal{C}$ to be the class of polynomials defined by $\mathcal{F}_V + \mathcal{F}_{OV}$ where $\mathcal{F}_{OV}$ is defined as in (3), i.e. uses sparse polynomials. This corresponds to the MQ-Sign-SS and MQ-Sign-RS variants. In this section we show that the usage of sparse $\mathcal{F}_{OV}$ introduces weaknesses that enable a practical key-recovery attack that takes merely seconds to mount. In the attack, we essentially solve the Extended Isomorphism of Polynomials (EIP) problem as defined in [4] (see also [13]). We recall here its definition.

$\mathsf{EIP}(n, m, \mathcal{P}, \mathcal{C})$:
Input: an m-tuple of multivariate polynomials $\mathcal{P} = (\mathcal{P}^{(1)}, \mathcal{P}^{(2)}, \ldots, \mathcal{P}^{(m)}) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$ and a special class of multivariate polynomial systems $\mathcal{C} \subseteq \mathbb{F}_q[x_1, \ldots, x_n]^m$.
Question: Find – if any – $S \in \mathrm{GL}_n(q)$ and $\mathcal{F} = (\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \ldots, \mathcal{F}^{(m)}) \in \mathcal{C}$ such that $\mathcal{P} = \mathcal{F} \circ S$.

Solving this problem is in general not easy, but if $\mathcal{F}$ exhibits enough structure, as is the case of MQ-Sign-SS and MQ-Sign-RS, then it becomes easy.

In order to see this, note that the computation of the public key for UOV-like signatures schemes can be written in matrix form as:

$$\begin{pmatrix} P_1^{(k)} & P_2^{(k)} \\ 0 & P_4^{(k)} \end{pmatrix} = \begin{pmatrix} I & 0 \\ S_1^\top & I \end{pmatrix} \begin{pmatrix} F_1^{(k)} & F_2^{(k)} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & S_1 \\ 0 & I \end{pmatrix}.$$

From this we deduce

$$\begin{pmatrix} P_1^{(k)} & P_2^{(k)} \\ 0 & P_4^{(k)} \end{pmatrix} = \begin{pmatrix} F_1^{(k)} & (F_1^{(k)} + F_1^{(k)\top})S_1 + F_2^{(k)} \\ 0 & \mathsf{Upper}(S_1^\top F_1^{(k)} S_1 + S_1^\top F_2^{(k)}) \end{pmatrix}.$$

From the two upper blocks we obtain the following two equations

$$\begin{aligned} P_1^{(k)} &= F_1^{(k)} \\ P_2^{(k)} &= (F_1^{(k)} + F_1^{(k)\top})S_1 + F_2^{(k)}. \end{aligned}$$

From these, we infer that

$$P_2^{(k)} = (P_1^{(k)} + P_1^{(k)\top})S_1 + F_2^{(k)}. \tag{4}$$

The matrices $F_2^{(k)}$ are part of the secret key, but we know that they are sparse. From the description of $\mathcal{F}_{OV}$ in (3) we can see that the value of $F_2^{(k)}$ is known on $(vm - v)$ entries. Since $F_2^{(k)}$ appears linearly in (4), we can extract constraints from the entries where the value of $F_2^{(k)}$ is zero. Let $\widetilde{P}_1 = P_1^{(k)} + P_1^{(k)\top}$. We obtain the following system of equations.

$$\sum_{1 \leqslant p \leqslant v} \widetilde{P}^{(k)}_{1[ip]} S_{1[pj]} - P^{(k)}_{2[ij]} = 0, \quad \forall (i,j,k) \text{ s.t. } F^{(k)}_{2[ij]} = 0. \tag{5}$$

This is a linear system in vm variables. The number of equations that we can obtain if we use all of the m quadratic maps from the public key is $mv(m-1)$. Hence, the system has vm linearly independent equations with overwhelming probability. As such, it can be solved efficiently through Gaussian Elimination.

Note from (5) that each equation in the system contains variables from only one column of S_1. This observation allows us to further optimize the attack by solving for one column at a time. Instead of solving one linear system in vm variables, we solve m linear systems in v variables. Thus, our attack has only $\mathcal{O}(mv^\omega)$ time complexity, where ω is the linear algebra constant. Table 1 summarizes the effect of the attack on the different MQ-Sign parameters.

Security level	Parameters (q, v, m)	Attack complexity
I	$(2^8, 72, 46)$	2^{24}
III	$(2^8, 112, 72)$	2^{27}
V	$(2^8, 148, 96)$	2^{29}

Table 1. Theoretical complexity of our attack against the MQ-Sign-SS and MQ-Sign-RS variants.

Our attack relies on two key properties. Firstly, we exploit the sparseness property of the vinegar-oil quadratic part. Secondly, we use the specific structure of the linear transformation S, as per the *equivalent keys* key generation technique.

4 A forgery attack on sparse $\mathcal{F}_V$

In this section we show a forgery attack on the MQ-Sign-SR variant, where the polynomials of $\mathcal{F}_V$ are defined as in Equation (3). A forgery attack on a multivariate signature scheme aims at finding a signature $\mathbf{z} \in \mathbb{F}^n$ for a given target value $\mathbf{t} \in \mathbb{F}^m$, such that $\mathcal{P}(\mathbf{z}) = \mathbf{t}$ is fulfilled. We show that in the case of MQ-Sign-SR, a forgery is directly possible using only the public key.

Recall from Section 3, that, when the linear transformation S is given as in Equation (2), it holds that $P_1^{(k)} = F_1^{(k)}$. This means that the sparsity of the

secret coefficient matrices $\mathcal{F}_V$ gets transferred to the public system. In more detail, an attacker faces the task of finding $(\mathbf{z}_v, \mathbf{z}_o) \in \mathbb{F}^n$ such that

$$(\mathbf{z}_v, \mathbf{z}_o) \begin{pmatrix} P_1^{(k)} & P_2^{(k)} \\ 0 & P_4^{(k)} \end{pmatrix} \begin{pmatrix} \mathbf{z}_v \\ \mathbf{z}_o \end{pmatrix} = \mathbf{z}_v P_1^{(k)} \mathbf{z}_v + \mathbf{z}_v P_2^{(k)} \mathbf{z}_o + \mathbf{z}_o P_4^{(k)} \mathbf{z}_o = t_k \tag{6}$$

holds for every $k \in \{1, \ldots, m\}$, where $P_1^{(k)}$ are sparse as in Equation 3. The parameters $n \approx 2.5m$ allow us to fix the m entries of $\mathbf{z}_o \in \mathbb{F}^m$ and thereby remove the non-sparse submatrices $P_2^{(k)}$ and $P_4^{(k)}$ from the quadratic part of this system of equations. This leads us to equations of the form

$$\mathbf{z}_v P_1^{(k)} \mathbf{z}_v + \mathrm{lin}(\mathbf{z}_v) = \sum_{i=1}^{v} \alpha_i^k z_i z_{(i+k-1(\bmod v))+1} + \mathrm{lin}(\mathbf{z}_v) = t_k. \tag{7}$$

The term $\mathrm{lin}(\mathbf{z}_v)$ summarizes the linear and constant terms emerging from Equation (6) after fixing the entries of $\mathbf{z}_o$. Note that the resulting system is a system of m equations in v variables, and since v is greater than m, we can still fix another $(v - m)$ variables and expect to have a solution.

At the core of this forgery attack is the observation that, due to the sparseness in $P_1^{(k)}$, the resulting system has subsets of equations that are bilinear in some subsets of variables. Specifically, upon closer examination of the indices in Equation (7), one notices that for odd k, the quadratic monomials appearing in the polynomial equation each consist of a variable with an odd and an even index. This implies that these $\frac{m}{2}$ equations are bilinear in the sets of variables $\{z_1, z_3, \ldots, z_{m-1}\}$ and $\{z_2, z_4, \ldots, z_m\}$, where we denote by z_i the variables in vector $\mathbf{z}_v$. Hence, randomly guessing e.g., the $\frac{v}{2}$ odd-indexed variables gives us a $\frac{v-m}{2}$-dimensional linear solution space for the even-indexed variables in the $\frac{m}{2}$ bilinear equations.

However, the probability that there exists a solution to the complete system - including the remaining $\frac{m}{2}$ quadratic (non-bilinear) equations - with the previously guessed odd variables is around $q^{-(\frac{v}{2}-(v-m))}$, since we can only fix $v - m$ variables in a quadratic system with v variables in m equations and still expect to find a solution. An alternative view is that, to obtain the $\frac{v-m}{2}$-dimensional linear solution space, we can fix $(v - m)$ variables and enumerate the rest with the usual cost of enumeration. This is the first step of our attack and its cost will be denoted by $C_{\mathrm{ENUM}(q, \frac{v}{2}-(v-m))}$.

In the second step, we need to find an assignment to the even-indexed variables that also validate the remaining $\frac{m}{2}$ equations. Using the description of the linear solution space obtained from the bilinear equations, this step boils down to solving a quadratic system of $\frac{m}{2}$ equations in $\frac{v-m}{2}$ variables. We denote the complexity of this step by $C_{\mathrm{MQ}(q, \frac{v-m}{2}, \frac{m}{2})}$. For the choice of $q = 2^8$, as per the MQ-Sign parameters, the best strategy would be to solve the system with a Gröbner-based algorithm (such as F4 or F5 [5, 6]), without the use of hybridization. Assuming that the quadratic systems we obtain behave as semi-regular non-boolean systems of s equations in n variables, the complexity [2] of

the solving algorithm is approximated by

$$\mathcal{O}\left(sD\binom{n+D-1}{D}^{\omega}\right),$$

where D denotes the *degree of regularity* and is computed as the power of the first non-positive coefficient in the expansion of

$$\frac{(1-t^2)^s}{(1-t)^n}.$$

Then, the complexity of the whole attack is given by

$$C_{\text{ENUM}(q,\frac{v}{2}-(v-m))} \cdot C_{\text{MQ}(q,\frac{v-m}{2},\frac{m}{2})},$$

since the second step has to be repeated until the odd variables are guessed correctly. In Table 2 we present an overview of the approximate costs for the parameter sets of MQ-Sign. We conclude that because of this attack, the proposed parameters of the MQ-Sign-SR variant slightly fail to provide the required security levels. Note that the algorithm described here uses the most straighforward approach to exploit the bilinearity of the subsystems, but more advanced techniques can potentially result in attacks with lower complexity.

Security level	Parameters (q, v, m)	$C_{\text{ENUM}(q,\frac{v}{2}-(v-m))}$	$C_{\text{MQ}(q,\frac{v-m}{2},\frac{m}{2})}$	Complexity
I	$(2^8, 72, 46)$	2^{80}	2^{31}	2^{111}
III	$(2^8, 112, 72)$	2^{128}	2^{42}	2^{170}
V	$(2^8, 148, 96)$	2^{176}	2^{52}	2^{228}

Table 2. Theoretical complexity of our forgery attack using the bilinear structure of the odd equations.

Our attack again relies on the sparseness property of the vinegar-vinegar quadratic part and the specific structure of the linear transformation S, as per the *equivalent keys* key generation technique.

5 Implementation

5.1 Sparse $\mathcal{F}_{OV}$

To confirm the practicality of our attack in Section 3, we provide a verification script in MAGMA [3] where we implement the key generation of MQ-Sign-{S/R}S and then run the main algorithm for recovering the secret key from the public key as input. The running time of the attack on a laptop is 0.6 seconds for the proposed parameters for security level I, 2.3 seconds for security level III and 6.9 seconds for security level V. We also provide an equivalent SageMath [14] script that is slower.

5.2 Sparse $\mathcal{F}_V$

Complexity estimates in Section 4 show that MQ-Sign-SR falls below the required security level, but the attack is not practical for the chosen parameter sizes. We nevertheless implemented the attack as a proof-of-concept and to confirm practically our complexity estimations. The cost of enumeration is straightforward, but the second part of the attack involves Gröbner-based algorithms, whose complexity rely on heuristic assumptions of semi-regularity. Hence, our primary goal in this experimental work was to verify that the degree of regularity reached by the F4/F5 algorithm is estimated correctly. The verification script for this attack consists of generating the polynomial system in (6), fixing all variables in $\mathbf{z}_o$ and in the odd-indexed subset, and finally, solving the resulting system using the F4 algorithm implemented in MAGMA. When fixing the variables, we experimented both with a correct assignment that subsequently leads to a solution, and a random assignment that leads to an inconsistent system. As expected, there is no difference in the solving running times between the two cases.

Security level	Parameters (q, v, m)	D estimated	D reached	Runtime (s)	Memory (MB)
I	$(2^8, 72, 46)$	4	4	0.6	32
III	$(2^8, 112, 72)$	5	5	90.2	534
V	$(2^8, 148, 96)$	6			> 32000

Table 3. Experimental results of the forgery attack.

The results of our experiments are in Table 3. Most notably, we confirm that the degree of regularity reached during the execution of the algorithm matches the theoretical estimation. This holds for both security level I and III. For security level V, the degree of regularity is expected to be six, hence we could not perform the verification due to the high memory requirements. For further assurance, we verified our complexity estimation on other parameter sets that are not part of the MQ-Sign specification, but follow the usual UOV ratios. We conclude that the MQ instances that need to be solved in the second part of the algorithm behave as semi-regular instances and the complexity of finding a solution can reliably be estimated using the analysis in [2].

Verification scripts for both attacks outlined in this paper can be found at

`https://github.com/mtrimoska/MQ-Sign-attack`.

6 Impact on the MQ-Sign variants

Both attacks presented in this paper rely on the specific structure of the linear transformation S, as per the *equivalent keys* key generation technique. This

technique is used in most modern UOV-based signature schemes, including MQ-Sign. If the equivalent keys structure is removed and S is a random affine map[4], this change of representation comes with additional memory cost. Specifically, Table 4 shows the impact of this modification on the secret key sizes, compared to the sizes reported in the MQ-Sign specification. The comparison is shown for the three MQ-Sign variants that are concerned by the two attacks proposed in this paper. The fourth variant, MQ-Sign-RR, is equivalent to the traditional UOV scheme and is not affected by our attacks. For this variant, the use of the equivalent keys structure of S is still a concern for side-channel attacks [10, 1].

Variant	Security Level					
	I		III		V	
	equivalent keys S	random S	equivalent keys S	random S	equivalent keys S	random S
MQ-Sign-SS	15561	26173	37729	63521	66421	111749
MQ-Sign-RS	133137	143749	485281	511073	1110709	1156037
MQ-Sign-SR	164601	175213	610273	636065	1416181	1461509

Table 4. Size (in Bytes) of the secret key of MQ-Sign with and without the equivalent keys structure of S.

Furthermore, this countermeasure was shown to be insufficient for the variants where the vinegar-oil space is sparse. In subsequent work, Ikematsu, Jo, and Yasuda [7] propose an attack that does not rely on the equivalent structure of S and remains practical: it runs in no more than 30 minutes for all security levels.

For the MQ-Sign-SR variant, further research is needed to determine whether the sparseness of $\mathcal{F}_V$ can still be exploited in a similar manner when S is random.

References

[1] T. Aulbach, F. Campos, J. Krämer, S. Samardjiska, and M. Stöttinger. Separating oil and vinegar with a single trace side-channel assisted Kipnis-Shamir attack on UOV. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):221–245, 2023.

[2] M. Bardet. *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université de Paris VI, 2004.

[3] W. Bosma, J. Cannon, and C. Playoust. The Magma Algebra System. I. The User Language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[4] This was suggested by the authors of MQ-Sign as a countermeasure when the attack in Section 3 was first announced.

[4] J. Ding, L. Hu, B.-Y. Yang, and J.-M. Chen. Note on Design Criteria for Rainbow-Type Multivariates. Cryptology ePrint Archive, Report 2006/307, 2006.

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

[6] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation ISSAC*, pages 75–83. ACM Press, 2002.

[7] Y. Ikematsu, H. Jo, and T. Yasuda. A security analysis on MQ-Sign. Cryptology ePrint Archive, Paper 2023/581, 2023. `https://eprint.iacr.org/2023/581`.

[8] A. Kipnis, J. Patarin, and L. Goubin. Unbalanced Oil and Vinegar Signature Schemes. In J. Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 206–222. Springer, 1999.

[9] A. Kipnis and A. Shamir. Cryptanalysis of the Oil & Vinegar Signature Scheme. In H. Krawczyk, editor, *CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 257–266. Springer, 1998.

[10] A. Park, K.-A. Shim, N. Koo, and D.-G. Han. Side-channel attacks on post-quantum signature schemes based on multivariate quadratic equations. 2018(3):500–523, 2018. `https://tches.iacr.org/index.php/TCHES/article/view/7284`.

[11] J. Patarin. The oil and vinegar signature scheme, 1997.

[12] A. Petzoldt. *Selecting and reducing key sizes for multivariate cryptography.* PhD thesis, Darmstadt University of Technology, Germany, 2013.

[13] K.-A. Shim1, J. Kim1, and Y. An. MQ-Sign: A New Post-Quantum Signature Scheme based on Multivariate Quadratic Equations: Shorter and Faster. https://www.kpqc.or.kr/images/pdf/MQ-Sign.pdf, 2022.

[14] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.5)*, 2022. `https://www.sagemath.org`.