Practical key-recovery attack on MQ-Sign and more

Thomas Aulbach[1], Simona Samardjiska[2], and Monika Trimoska[3]*

[1] University of Regensburg, Regensburg, Germany
[2] Radboud Universiteit, Nijmegen, The Netherlands
[3] Eindhoven University of Technology, Eindhoven, The Netherlands
thomas.aulbach@ur.de,simonas@cs.ru.nl,m.trimoska@tue.nl

Abstract. In this paper we describe attacks on the UOV-based signature scheme called MQ-Sign. MQ-Sign was submitted by Shim, Kim, and An as a a first-round candidate for standardization in the (South) Korean post-quantum cryptography competition (KpqC). The scheme makes use of sparseness of the secret central polynomials and equivalent key construction to reduce the size of the private key. The authors propose four variants exploiting different levels of sparsity, MQ-Sign-SS, MQ-Sign-RS, MQ-Sign-SR, and MQ-Sign-RR with the last one being the standard UOV signature scheme.

We show that apart from the MQ-Sign-RR variant, all the others are insecure. Namely, we present a polynomial-time key-recovery attack on the variants MQ-Sign-SS and MQ-Sign-RS and a forgery attack on the variant MQ-Sign-SR below the claimed security level. Our attack exploits exactly the techniques used for reduction of keys - the sparsity of the central polynomials in combination with the specific structure of the secret linear map $\mathbf{S}$.

We provide a verification script for the polynomial-time key-recovery attack, that recovers the secret key in less than seven seconds for security level V. Furthermore, we provide an implementation of the non-guessing part of the forgery attack, confirming our complexity estimates.

1 Introduction

In recent years we have witnessed a substantial effort from standardization bodies and the cryptographic community to design, develop and scrutinize candidates for post-quantum secure key-encapsulation mechanisms and digital signatures [20, 12, 14, 26, 7]. This effort is racing an equally fuelled one for developing a large scale error-tolerant universal quantum computer which, although still very much elusive, will likely be reality in a decade or so [19]. When this happens, all the classical cryptography we are happily using today will be immediately rendered insecure. Therefore, as the community widely agrees upon, we need to

* This work was partially done and published for the first time while the author was at Radboud University.

move as fast as possible with the standardization of post-quantum cryptosystems that we believe are secure even against quantum adversaries.

On the other hand, we need to be extremely careful in the assessment of the level of scrutiny put into these standardization processes. For example, a major disruption in NIST's standardization process, and certainly a shock for the crypto community, was the cryptanalysis [28, 3] of the two multivariate quadratic ($\mathcal{MQ}$) signature schemes - GeMSS [6] and Rainbow [8] after they were chosen as finalists [20]. Both of these schemes were thought to be well understood, with solid security analysis, albeit both with ad-hoc designs and no security proof.

These developments resulted in NIST choosing two lattice-based signatures schemes in the new standard [25, 18] in addition to the heavy SPHINCS+ [13], and no adequate solution for use-cases in need of very small signatures. NIST reopened the call for post-quantum digital signature proposals, specifying the need for shorter signatures with fast verification. This spurred a huge number of new multivariate signatures many of which variants of UOV (Unbalanced Oil and Vinegar) [16]. UOV is one of the oldest, simplest and most studied ad-hoc multivariate signatures schemes. It has very short signatures, but the public key is huge. Therefore, it was not particularly interesting for a very long time, especially since the alternative Rainbow seemed to be more efficient for the same security level (after the attack by Beullens [3] this advantage disappeared). After Rainbow was out of the game, the community returned to UOV in a new round of attempts to reduce the size of the public key while not compromising the security.

One of those efforts is the MQ-Sign [27] signature scheme submitted to the Korean Post-Quantum Cryptography Competition [26], and since recently selected to advance to the 2nd round. The MQ-Sign submission combines two known techniques from multivariate cryptography - equivalent keys [24] and sparse central polynomials [30]. The central map is a standard UOV map that can additionally exhibit sparseness in either the vinegar-vinegar part or the vinegar-oil part. The authors propose four different variants: Both the vinegar-vinegar and vinegar-oil parts being sparse corresponds to the MQ-Sign-SS variant, which yields the smallest private keys. In the variant MQ-Sign-RS, the vinegar-vinegar part is random and the vinegar-oil part is sparse. The two parts switch their structure in the MQ-Sign-SR variant. Finally, the variant MQ-Sign-RR, where both parts are random, corresponds to the standard UOV signature scheme.

1.1 Our contribution

In this work, we study the security of the MQ-Sign signature scheme. We propose two attacks that cover all variants using sparseness, i.e. every except the last, MQ-Sign-RR variant.

First, 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-SS and MQ-Sign-RS. Our attack relies on two key properties – the sparseness property of the vinegar-oil quadratic part and the specific structure of the linear

transformation $\mathbf{S}$, as per the *equivalent keys* key generation technique. We first recover the linear transformation $\mathcal{S}$, which allows to subsequently compute the central map $\mathcal{F}$. Our attack is very efficient, and recovers the key in just seconds regardless of the security level.

Second, we introduce a forgery attack on the variant MQ-Sign-SR which is actually a direct attack using only the public key. Our attack exploits a bilinear substructure emerging as a result of the sparse secret polynomials. The attack is not practical, but still shows that MQ-Sign-SR falls short of the claimed security levels by about 30 bits.

We perform a complexity analysis of both attacks, showing that these three variants do not reach the originally estimated security levels. The claims in our complexity analysis are additionally backed up with experimental results. Most notably, we provide an implementation of the practical key-recovery attack that is executed in less than seven seconds for all security levels. We also provide an implementation of the non-guessing part of the forgery attack, confirming our complexity estimates. Both the implementation of attacks and the code used for confirming the complexity estimates are open source.

1.2 Timeline

Our key recovery attack on MQ-Sign-RS and MQ-Sign-SS with $\mathcal{S}$ in block matrix structure (using the equivalent keys optimization) was announced in March 2023. Shortly afterwards, Ikematsu, Jo, and Yasuda [15] generalized our approach and gave an efficient attack that also works with general $\mathcal{S}$. As a result of the two attacks, the authors of MQ-Sign removed the two variants MQ-Sign-RS and MQ-Sign-SS from their specifications in the ongoing KpqC competition. Note that in the current version of the specifications, both remaining variants still use the equivalent key optimization, and do not use a random linear transformation $\mathcal{S}$.

1.3 Organization of the paper

In Section 2 we provide the necessary background on multivariate cryptography, in particular the UOV signature scheme and the optimization choices used in MQ-Sign. We introduce the announced attacks in Section 3 and 4. In more detail, we first show in Section 3 that the sparse vinegar-oil polynomials in MQ-Sign-RS and MQ-Sign-SS let us derive enough linear equations to compute the secret linear transformation $\mathcal{S}$ in a matter of seconds. Section 4 demonstrates a strategy to attack MQ-Sign-SR by first guessing a selection of variables and subsequently solving a part of the equations for the remaining ones. Even though the cost of the guessing part remains quite high, this shows that the remaining sparse variant slightly fails to provide the required security levels. We provide verification scripts of the stated attacks in Section 5 and discuss the impact on the MQ-Sign variants in Section 6. Finally, we debate about the still appealing question of using sparse polynomials in UOV and shift attention to the public equations instead.

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 $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$.

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 $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}_q^m$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}_q^m$ and computes recursively $\mathbf{x} = \mathcal{T}^{-1}(\mathbf{w}) \in \mathbb{F}_q^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}_q^n$, and $\mathbf{z} = \mathcal{S}^{-1}(\mathbf{y})$. The signature of the message d is $\mathbf{z} \in \mathbb{F}_q^n$. 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 $\mathbf{z} \in \mathbb{F}_q^n$ is indeed a valid signature for a message d, one computes $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}_q^m$. 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.

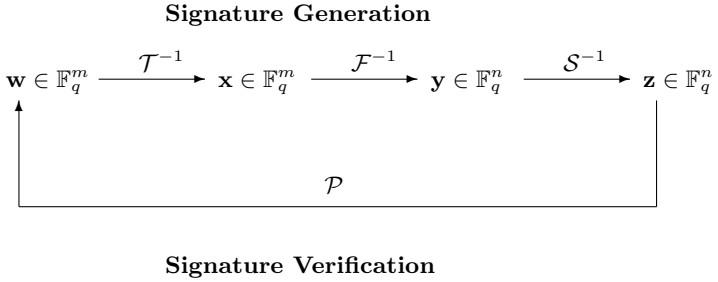

Fig. 1. General workflow of multivariate signature schemes.

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, Patarin, and Goubin at EUROCRYPT'99 [16] as a modification of the oil and vinegar scheme of Patarin [22] that was broken by Kipnis and Shamir in 1998 [17].

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 [23] that for any instance of a UOV secret key $(\mathcal{F}, \mathcal{S})$, there exists an equivalent secret key $(\mathcal{F}, \mathbf{S})$ with

$$\mathbf{S} = \begin{pmatrix} \mathbf{I}_{v\times v} & \mathbf{S}_1 \\ \mathbf{0}_{m\times v} & \mathbf{I}_{m\times m} \end{pmatrix}. \tag{2}$$

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 $\mathbf{F}^{(1)}, \ldots, \mathbf{F}^{(m)} \in \mathbb{F}_q^{n\times n}$ where each nonzero coefficient (i, j) in $\mathbf{F}^{(k)}$ 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 $\mathbf{F}_1^{(k)} \in \mathbb{F}_q^{v\times v}$ and a block $\mathbf{F}_2^{(k)} \in \mathbb{F}_q^{v\times m}$ on the top right. In other words, the matrices are of the form:

$$\mathbf{F}^{(k)} = \begin{pmatrix} \mathbf{F}_1^{(k)} & \mathbf{F}_2^{(k)} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Thus, in order to obtain a key pair, it suffices to first randomly generate $(\mathbf{S}_1, \mathbf{F}^{(1)}, \ldots, \mathbf{F}^{(m)})$ and then compute $(\mathbf{P}^{(1)}, \ldots, \mathbf{P}^{(m)})$ by evaluating $\mathbf{P}^{(k)} = \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S}$ 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

$$\mathcal{F}_V^{(k)}(x_1, \ldots, x_n) = \sum_{i\in V, j\in V} \gamma_{ij}^{(k)} x_i x_j, \text{ and } \quad \mathcal{F}_{OV}^{(k)}(x_1, \ldots, x_n) = \sum_{i\in V, j\in O} \gamma_{ij}^{(k)} x_i x_j.$$

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 main design goal of MQ-Sign is to reduce the size of the secret key compared to traditional UOV. This is achieved using sparse polynomials for the quadratic part of the central map. If sparseness is introduced in the $\mathcal{F}_V^{(k)}$ part, then it is defined as

$$\mathcal{F}_{V,S}^{(k)}(x_1, \ldots, x_n) = \sum_{i=1}^{v} \alpha_i^k x_i x_{(i+k-1(\bmod v))+1} \tag{3}$$

If, on the other hand, sparseness is introduced in the $\mathcal{F}_{OV}^{(k)}$ part, then it is defined as

$$\mathcal{F}_{OV,S}^{(k)}(x_1,\ldots,x_n) = \sum_{i=1}^{v} \beta_i^k x_i x_{(i+k-2(\bmod m))+v+1}. \tag{4}$$

The MQ-Sign proposal provides a parameter selection for four variants 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 ($\mathcal{F}_{V,S}$) or random polynomials ($\mathcal{F}_{V,R}$). The second S/R refers to the same property, but for $\mathcal{F}_{OV}$. Note that the variant MQ-Sign-RR corresponds to the standard UOV scheme defined with inhomogenous polynomials.

If both $\mathcal{F}_{V,S}$ and $\mathcal{F}_{OV,S}$ are used, the size of the secret key is reduced to $2vm$ field elements.

The authors provide an elaborate security analysis including all known relevant attacks on UOV. However, they do not consider the sparseness of (parts of) the secret polynomials in any of the attacks. Their assumption is that it is not exploitable within the known cryptanalytic techniques. Table 1 summarizes the parameters chosen by the authors for security levels I, III, and V.

Note that when $\mathcal{F}_{V,S}$ is used, the size of the public key can also be reduced, as, due to the equivalent keys structure of $\mathbf{S}$ as in (2), a part of the public key is equivalent to a part of the secret key and thus sparse. This is however not taken into consideration in the implementation of MQ-Sign or in the public key sizes reported in Table 1.

Sec. level	Parameters (q,v,m)	sig	PK	SK (SS)	SK (RS)	SK (SR)	SK (RR)
I	$(2^8,72,46)$	134	328 441	15 561	133 137	164 601	282 177
III	$(2^8,112,72)$	200	1 238 761	37 729	485 281	610 273	1 057 825
V	$(2^8,148,96)$	260	2 892 961	66 421	1 110 709	1 416 181	2 460 469

Table 1. The parameter selection for security category I, III and V for the variants SS, RS, SR and RR of MQ-Sign with key sizes in bytes.

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

In the following, we consider $\mathcal{C}$ to be the class of polynomials defined by $\mathcal{F}_{V,R} + \mathcal{F}_{OV,S}$, denoting that only $\mathcal{F}_{OV}$ needs to be defined as in (4), i.e. with sparse polynomials. This corresponds to the MQ-Sign-SS and MQ-Sign-RS variants.

In this section we show that the usage of $\mathcal{F}_{OV,S}$ 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 [9] (see also [27]). 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 m-tuples of multivariate polynomials $\mathcal{C} \subseteq \mathbb{F}_q[x_1, \ldots, x_n]^m$.
Question: Find – if any – $\mathbf{S} \in \mathrm{GL}_n(q)$ and $\mathcal{F} = (\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, \ldots, \mathcal{F}^{(m)}) \in \mathcal{C}$ such that $\mathcal{P} = \mathcal{F} \circ \mathbf{S}$.

Solving this problem is in general not easy. In fact, the security of ad-hoc multivariate schemes is based on the hardness on this problem. However, if $\mathcal{F}$ exhibits enough structure, then the problem can become easy to solve.

We next show that the sparse structure present in MQ-Sign-SS and MQ-Sign-RS is enough to solve the corresponding EIP problem very efficiently. 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:

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ \mathbf{0} & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{S}_1^\top & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{F}_1^{(k)} & \mathbf{F}_2^{(k)} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{S}_1 \\ \mathbf{0} & \mathbf{I} \end{pmatrix}.$$

From this we deduce

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ \mathbf{0} & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1^{(k)} & (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)} \\ \mathbf{0} & \mathsf{Upper}(\mathbf{S}_1^\top \mathbf{F}_1^{(k)} \mathbf{S}_1 + \mathbf{S}_1^\top \mathbf{F}_2^{(k)}) \end{pmatrix}. \tag{5}$$

Equation (5) shows how different blocks from the public key are obtained from the blocks of the secret key, and having these relations allows us to mount an algebraic attack that will recover all of the entries of the secret key. We first modelize this correspondence between the public and the secret key as a system of equations where the variables are the entries of $\mathbf{S}_1$ and $\mathbf{F}_1^{(k)}$. From the two upper blocks we obtain the following two equations

$$\begin{aligned} \mathbf{P}_1^{(k)} &= \mathbf{F}_1^{(k)} \\ \mathbf{P}_2^{(k)} &= (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)}. \end{aligned}$$

From these, we infer that

$$\mathbf{P}_2^{(k)} = (\mathbf{P}_1^{(k)} + \mathbf{P}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)}. \tag{6}$$

Ignoring the sparseness at first, from (6) we can derive a linear system of vm^2 equations in $v(m^2 + m)$ variables (vm that correspond to the entries of the unknown block of the linear transformation $\mathbf{S}$, and vm^2 from the entries of $\mathbf{F}_2^{(k)}$). Even though the system is linear, a solution can not be extracted easily as it is highly underdetermined. But considering the sparseness in the MQ-Sign-SS and MQ-Sign-RS instances, the following key observation allows us to solve the system easily in practice.

The matrices $\mathbf{F}_2^{(k)}$ are part of the secret key, but we know that they are sparse. From the description of $\mathcal{F}_{OV}$ in (4) we can see that the value of $\mathbf{F}_2^{(k)}$ is known on

$(vm - v)$ entries. Since $\mathbf{F}_2^{(k)}$ appears linearly in (6), we can extract constraints from the entries where the value of $\mathbf{F}_2^{(k)}$ is zero and obtain a system that is only in the $\mathbf{S}_1$ variables. Let $\widetilde{\mathbf{P}}_1^{(k)} = \mathbf{P}_1^{(k)} + \mathbf{P}_1^{(k)\top}$. We obtain the following system of equations, where we denote by $\tilde{p}_{i,j}^{(k)}$ the entries of $\widetilde{\mathbf{P}}_1^{(k)}$, by $s_{i,j}$ the entries of $\mathbf{S}_1$, by $p_{i,j}^{(k)}$ the entries of $\mathbf{P}_2^{(k)}$, and by $f_{i,j}^{(k)}$ the entries of $\mathbf{F}_2^{(k)}$.[4]

$$\sum_{1 \leqslant p \leqslant v} \tilde{p}_{i,p}^{(k)} s_{p,j} - p_{i,j}^{(k)} = 0, \quad \forall (i,j,k) \text{ s.t. } f_{i,j}^{(k)} = 0. \tag{7}$$

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. This is under the assumption that the system behaves as a random system and has no specific structure that results in non-trivial dependencies between the equations, which will be argumented below as part of the complexity analysis. We conclude that, ignoring some of the equations from (6), specifically those where $f_{i,j}^{(k)}$ is not zero, allowed us to derive a linear system that is only in variables from $\mathbf{S}_1$. Once we recover the secret map $\mathbf{S}$, computing $\mathcal{F}$ is easy, as we just need to apply the inverse linear transformation on $\mathcal{P}$.

We further refine our modelisation to obtain a more efficient attack, using the following strategy. Note from (7) that each equation in the system contains variables from only one column of $\mathbf{S}_1$. This observation allows us to optimize the attack by solving for one column at a time. This is more evident when we look at the matrix representation of our linear system. Let us define a matrix $\mathbf{A}'$ as

$$\begin{pmatrix} \widetilde{\mathbf{P}}_1^{(1)} \\ \widetilde{\mathbf{P}}_1^{(2)} \\ \dots \\ \widetilde{\mathbf{P}}_1^{(m)} \end{pmatrix},$$

i.e. a block matrix obtained by concatenating vertically the quadratic maps $\widetilde{\mathbf{P}}_1^{(k)}$. Then, let $\mathbf{A}$ be a block matrix that has copies of $\mathbf{A}'$ on the main diagonal and zeros everywhere else

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}' & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}' & \dots & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}' \end{pmatrix}.$$

[4] Here, and in the following, the submatrix indices are ommited where there is no ambiguity

Now, let $\mathbf{x}^\top = \begin{pmatrix}\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_m\end{pmatrix}$ be a vector obtained by concatenating the columns of $\mathbf{S}_1$. Finally, let $\mathbf{b}^\top = \begin{pmatrix}\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_m\end{pmatrix}$ be a vector that is obtained by concatenating the first column of each quadratic map $\mathbf{P}_2^{(k)}$, followed by the second column of each map, etc.

We can then rewrite $\widetilde{\mathbf{P}}_1^{(k)}\mathbf{S}_1 = \mathbf{P}_2^{(k)}$, for all $k \in \{1, \dots, m\}$, as $\mathbf{A}\mathbf{x} = \mathbf{b}$. Indeed, we have

$$\begin{pmatrix} \mathbf{A}' & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{A}' & \dots & \mathbf{0} \\ & & \ddots & \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{A}' \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_m \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \\ \mathbf{b}_2 \\ \\ \dots \\ \\ \mathbf{b}_m \\ \\ \end{pmatrix},$$

where

$$\mathbf{A}' = \begin{pmatrix} \tilde{p}_{1,1}^{(1)} & \dots & \tilde{p}_{1,v}^{(1)} \\ \tilde{p}_{2,1}^{(1)} & \dots & \tilde{p}_{2,v}^{(1)} \\ & \dots & \\ \tilde{p}_{v,1}^{(1)} & \dots & \tilde{p}_{v,v}^{(1)} \\ \\ & \dots & \\ \\ \tilde{p}_{1,1}^{(m)} & \dots & \tilde{p}_{1,v}^{(m)} \\ \tilde{p}_{2,1}^{(m)} & \dots & \tilde{p}_{2,v}^{(m)} \\ & \dots & \\ \tilde{p}_{v,1}^{(m)} & \dots & \tilde{p}_{v,v}^{(m)} \end{pmatrix}, \quad \mathbf{x}_i = \begin{pmatrix} s_{1,i} \\ s_{2,i} \\ \dots \\ s_{v,i} \end{pmatrix}, \quad \text{and} \quad \mathbf{b}_i = \begin{pmatrix} p_{1,i}^{(1)} \\ p_{2,i}^{(1)} \\ \dots \\ p_{v,i}^{(1)} \\ \\ \dots \\ \\ p_{1,i}^{(m)} \\ p_{2,i}^{(m)} \\ \dots \\ p_{v,i}^{(m)} \end{pmatrix}.$$

Looking at where the zero entries lie in $\mathbf{A}$, we can now split the problem. We solve $\mathbf{A}'\mathbf{x}_i = \mathbf{b}_i$ for all $i \in \{1, \dots, m\}$, and for every system that we solve, we reveal one column of $\mathbf{S}_1$.

3.1 Complexity analysis

Using this strategy, 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. A strong requirement for the success of the attack is that all of the linear subsystems that we need to

solve are determined. Since we are combining solutions of subsystems to recover the entire solution, having even a small nonzero number of solutions to the subsystems would rapidly increase the complexity of the attack. However, in the following, we argue that we can rely on the assumption that all subsystems have exactly one solution.

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

As per the analysis in the previous section, the ith subset of equations is obtained from

$$\begin{pmatrix} \tilde{p}_{1,1}^{(1)} & \dots & \tilde{p}_{1,v}^{(1)} \\ \tilde{p}_{2,1}^{(1)} & \dots & \tilde{p}_{2,v}^{(1)} \\ & \dots & \\ \tilde{p}_{v,1}^{(1)} & \dots & \tilde{p}_{v,v}^{(1)} \\ & \dots & \\ \tilde{p}_{1,1}^{(m)} & \dots & \tilde{p}_{1,v}^{(m)} \\ \tilde{p}_{2,1}^{(m)} & \dots & \tilde{p}_{2,v}^{(m)} \\ & \dots & \\ \tilde{p}_{v,1}^{(m)} & \dots & \tilde{p}_{v,v}^{(m)} \end{pmatrix} \cdot \begin{pmatrix} s_{1,i} \\ s_{2,i} \\ \dots \\ s_{v,i} \end{pmatrix} = \begin{pmatrix} p_{1,1}^{(1)} \\ p_{2,1}^{(1)} \\ \dots \\ p_{v,1}^{(1)} \\ \\ \dots \\ \\ p_{1,1}^{(m)} \\ p_{2,1}^{(m)} \\ \dots \\ p_{v,1}^{(m)} \end{pmatrix}. \tag{8}$$

From this equality, we extract $v(m-1)$ equations. That is, one equation for each entry from $\mathbf{b}_i$, ignoring entries (i, j) where $f_{i,j}^{(k)}$ is not zero. We are interested in how many of these equations are linearly independent. From (8) we can see that each equation can be viewed as a linear combination of the $s_{\_,i}$ variables where the coefficients come from a row of $\widetilde{\mathbf{P}}_1^{(k)}$, plus a constant that corresponds to an entry of $\mathbf{P}_2^{(k)}$. Hence, the number of linearly independent equations is exactly determined by the rank of $\mathbf{A}'$. It is actually the rank of $\left(\mathbf{A}' \; \mathbf{b}_i\right)$, but we can ignore the constant in our case. Indeed, if the rank of $\mathbf{A}'$ is smaller than the rank of $\left(\mathbf{A}' \; \mathbf{b}_i\right)$, this would result in the system derived from (8) being inconsistent. This case can not happen when we model a coherent instance of UOV key generation. Now, recall that public key in UOV-based schemes is generated randomly (or derived from a randomly generated central map) and

thus it is comprised of matrices of full rank with high probability. Hence, a concatenation of several such matrices is also full rank, which is v in this case (the dimension of the column space being v) – equal to the number of variables. We have also performed experiments to verify this claim, and out of 500 runs of the attack on MQ-Sign-SS with level I parameters, not once did the attack fail for not having enough independent equations in any of the subsystems. Table 2 summarizes the effect of the attack on the different MQ-Sign parameters.

4 A forgery attack on variants using 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 $\mathbf{z} \in \mathbb{F}_q^n$ for a given target value $\mathbf{t} \in \mathbb{F}_q^m$, 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 $\mathcal{S}$ is given as in Equation (2), it holds that $\mathbf{P}_1^{(k)} = \mathbf{F}_1^{(k)}$. This means that the sparsity of the secret coefficient matrices gets transferred to the public system. In more detail, an attacker faces the task of finding $(\mathbf{z}_v, \mathbf{z}_o) \in \mathbb{F}_q^n$ such that

$$(\mathbf{z}_v, \mathbf{z}_o) \begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ \mathbf{0} & \mathbf{P}_4^{(k)} \end{pmatrix} \begin{pmatrix} \mathbf{z}_v \\ \mathbf{z}_o \end{pmatrix} = \mathbf{z}_v \mathbf{P}_1^{(k)} \mathbf{z}_v + \mathbf{z}_v \mathbf{P}_2^{(k)} \mathbf{z}_o + \mathbf{z}_o \mathbf{P}_4^{(k)} \mathbf{z}_o = t_k \tag{9}$$

holds for all $k \in \{1, \ldots, m\}$, where $\mathbf{P}_1^{(k)}$ are sparse as in Equation (3). The parameters $n \approx 2.5m$ allow us to fix the m entries of $\mathbf{z}_o \in \mathbb{F}_q^m$ and thereby remove the non-sparse submatrices $\mathbf{P}_2^{(k)}$ and $\mathbf{P}_4^{(k)}$ from the quadratic part of this system of equations. This leads us to equations of the form

$$\mathbf{z}_v \mathbf{P}_1^{(k)} \mathbf{z}_v + \mathrm{lin}(\mathbf{z}_v) = \sum_{i=1}^{v} \alpha_i^k z_i z_{(i+k-1(\bmod v))+1} + \mathrm{lin}(\mathbf{z}_v) = t_k. \tag{10}$$

The term $\mathrm{lin}(\mathbf{z}_v)$ summarizes the linear and constant terms emerging from Equation (9) 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 fix another $(v - m)$ variables and still expect to have a solution.

At the core of this attack is the observation that, due to the sparsity in $\mathbf{P}_1^{(k)}$, the resulting system has subsets of equations that are bilinear in some subsets of variables. Specifically, upon closer examination of the indices in Equation (10), 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.

Let us denote by $\tilde{\mathbf{z}}_v$ the vector comprised of the vinegar variables that have not yet been assigned, i.e. the even-indexed vinegar variables. At this point, the overall system is of the following form

$$\sum_{i=0}^{\frac{v}{2}-1} \alpha_{2i+1}^k z_{((2i+1)+k-1(\bmod v))+1} + \alpha_{2i+2}^k z_{2i+2} + \mathrm{lin}(\tilde{\mathbf{z}}_v) = t_k, \qquad \text{if } k \text{ odd}$$

$$\sum_{i=1}^{\frac{v}{2}} \alpha_{2i}^k z_{2i} z_{(2i+k-1(\bmod v))+1} + \mathrm{lin}(\tilde{\mathbf{z}}_v) = t_k, \qquad \text{if } k \text{ even.}$$

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 $C_{\mathrm{MQ}(q,\frac{v-m}{2},\frac{m}{2})}$.

4.1 Complexity analysis

The cost of the first step of the algorithm corresponds to the usual cost of enumeration over $\mathbb{F}_q$. In the second step, the complexity is dominated by the algorithm for solving the quadratic systems of equations. 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 [10, 11]), 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

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

since the second step has to be repeated until the odd variables are guessed correctly in the first step. In Table 3 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 3. Theoretical complexity of our direct 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 $\mathbf{S}$, 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 [5] 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 [29] 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 (9), 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 4. Experimental results of the direct attack.

The results of our experiments are in Table 4. 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 $\mathbf{S}$, 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 $\mathbf{S}$ is a random affine map[5], this change of representation comes with additional memory cost. Specifically, Table 5 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 $\mathbf{S}$ is still a concern for side-channel attacks [21, 1].

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

Variant	Security Level					
	I		III		V	
	equivalent keys $\mathbf{S}$	random $\mathbf{S}$	equivalent keys $\mathbf{S}$	random $\mathbf{S}$	equivalent keys $\mathbf{S}$	random $\mathbf{S}$
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 5. Size (in Bytes) of the secret key of MQ-Sign with and without the equivalent keys structure of $\mathbf{S}$.

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 [15] propose an attack that does not rely on the equivalent structure of $\mathbf{S}$ 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 $\mathbf{S}$ is random.

7 Discussion on using sparse matrices

MQ-Sign follows the UOV construction that is widely believed to be solid. Yet, as we have demonstrated, bad choices for optimization have significantly damaged its security. The aforementioned attacks were possible due to mainly two reasons. First, the *secret* polynomials were chosen sparse. Thus, we could derive more equations from the public key entries and their computation in Equation (6) than there are secret key entries to obscure them. Second, the secret key polynomials were chosen so sparse, that half of the public key equations turned bilinear after fixing certain variables. The question that remains is whether we can still make use of sparseness to reduce the size of the (expanded) keys.

As an alternative, we could, instead of choosing sparse secret submatrices $\mathbf{F}_1^{(k)}$ and $\mathbf{F}_2^{(k)}$, choose the public $\mathbf{P}_1^{(k)}$ and $\mathbf{P}_2^{(k)}$ sparse. Our key-recovery attack does not work anymore, but, we would need to add more coefficients to the matrices, so that the strategy in Section 4 does not apply anymore.

The approach complements current UOV instantiations [4] which use key compression techniques. The authors of [4] expand the matrices $\mathbf{P}_1^{(k)}$ and $\mathbf{P}_2^{(k)}$ from a seed $\mathtt{seed}_{pk}$ and only store $cpk = (\mathtt{seed}_{pk}, \mathbf{P}_3^{(k)})$. Therefore, making these two matrices sparse will not result in a smaller compressed public key, but the size of the expanded secret key and the expanded public key would be reduced, which implies a lower overall storage requirement.

However, caution should be put into the choice of the sparse public matrices. The strategy of using "rotating diagonals" seems to work well with regards to the standard attacks against UOV analyzed in the specs. However, the sparse equations introduce enough structure to make a direct attack cheaper than in

the no-sparse case. An option could be to slightly increase the number of non-zero coefficients in $\mathbf{P}_1^{(k)}$ and $\mathbf{P}_2^{(k)}$, enough to increase the cost of our attack or a similar direct attack. This is of course an ad-hoc solution, and more scrutiny is required in order to determine whether a secure balance can be found that is a better solution than simply increasing the parameters. We leave this question as future work.

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. Beullens. Breaking Rainbow Takes a Weekend on a Laptop. In *CRYPTO*, volume 13508 of *Lecture Notes in Computer Science*, pages 464–479. Springer, 2022.

[4] W. Beullens, M.-S. Chen, S.-H. Hung, M. J. Kannwischer, B.-Y. Peng, C.-J. Shih, and B.-Y. Yang. Oil and vinegar: Modern parameters and implementations. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 321–365, 2023.

[5] 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).

[6] A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, and J. Ryckeghem. GeMSS. Technical report, National Institute of Standards and Technology, 2020.

[7] Chinese Association for Cryptologic Research (CACR). CACR post-quantum competition, 2018.

[8] J. Ding, M.-S. Chen, A. Petzoldt, D. Schmidt, B.-Y. Yang, M. Kannwischer, and J. Patarin. Rainbow. Technical report, National Institute of Standards and Technology, 2020.

[9] 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.

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

[11] 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.

[12] I. O. for Standardization ISO/IEC JTC 1/SC 27 (WG2). Information security, cybersecurity and privacy protection: ISO/IEC WD 14888-4 Infor-

mation technology — Security techniques — Digital signatures with appendix — Part 4: Stateful hash-based mechanisms. https://www.iso.org/standard/80492.html.

[13] A. Hulsing, D. J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, P. Kampanakis, S. Kolbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe, J.-P. Aumasson, B. Westerbaan, and W. Beullens. SPHINCS+. NIST PQC Submission, 2020.

[14] A. Hülsing, D. Butin, S.-L. Gazdag, J. Rijneveld, and A. Mohaisen. XMSS: extended hash-based signatures. RFC 8391, 2018.

[15] Y. Ikematsu, H. Jo, and T. Yasuda. A security analysis on MQ-Sign. In H. Kim and J. Youn, editors, *Information Security Applications*, pages 40–51, Singapore, 2024. Springer Nature Singapore.

[16] 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.

[17] 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.

[18] V. Lyubashevsky, L. Ducas, E. Kiltz, T. Lepoint, P. Schwabe, G. Seiler, D. Stehlé, and S. Bai. CRYSTALS-DILITHIUM. NIST PQC Submission, 2020.

[19] M. Mosca and M. Piani. 2021 Quantum Threat Timeline Report, 2022.

[20] National Institute for Standards and Technology. Post-Quantum Cryptography Standardization, 2017.

[21] 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.

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

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

[24] A. Petzoldt, S. Bulygin, and J. Buchmann. CyclicRainbow - A multivariate Signature Scheme with a Partially Cyclic Public Key based on Rainbow. Cryptology ePrint Archive, Report 2010/424, 2010.

[25] T. Prest, P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang. FALCON. NIST PQC Submission, 2020.

[26] Quantum resistant cryptography research center. Korean post-quantum cryptographic competition, 2022.

[27] K.-A. Shim, J. Kim, 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.

[28] C. Tao, A. Petzoldt, and J. Ding. Efficient Key Recovery for All HFE Signature Variants. In *CRYPTO (1)*, volume 12825 of *Lecture Notes in Computer Science*, pages 70–93. Springer, 2021.

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

[30] B.-Y. Yang, J.-M. Chen, and Y.-H. Chen. TTS: High-Speed Signatures on a Low-Cost Smart Card. In *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 371–385. Springer, 2004.