MAYO: Practical Post-quantum Signatures from Oil-and-Vinegar Maps

Ward Beullens(✉)

imec-COSIC, KU Leuven, Leuven, Belgium
Ward.beullens@esat.kuleuven.be

Abstract. The Oil and Vinegar signature scheme, proposed in 1997 by Patarin, is one of the oldest and best understood multivariate quadratic signature schemes. It has excellent performance and signature sizes but suffers from large key sizes on the order of 50 KB, which makes it less practical as a general-purpose signature scheme. To solve this problem, this paper proposes MAYO, a variant of the UOV signature scheme whose public keys are two orders of magnitude smaller. MAYO works by using a UOV map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ with an unusually small oil space, which makes it possible to represent the public key very compactly. The usual UOV signing algorithm fails if the oil space is too small, but MAYO works around this problem by "whipping up" the oil and vinegar map $\mathcal{P}$ into a larger map $\mathcal{P}^\star : \mathbb{F}_q^{kn} \to \mathbb{F}_q^m$, that does have a sufficiently large oil space. With parameters targeting NISTPQC security level I, MAYO has a public key size of only 614 Bytes and a signature size of 392 Bytes. This makes MAYO more compact than state-of-the-art lattice-based signature schemes such as Falcon and Dilithium. Moreover, we can choose MAYO parameters such that, unlike traditional UOV signatures, signatures provably only leak a negligible amount of information about the private key.

1 Introduction

The Oil and Vinegar signature scheme, introduced by Patarin in 1997, is a simple and seemingly well understood signature scheme in Multivariate Quadratic (MQ) cryptography. This scheme is based on a trapdoored multivariate map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, which consists of m multivariate quadratic polynomials in n variables. The trapdoor is a secret m-dimensional linear subspace O of $\mathbb{F}_q^n$, called the *oil space*, on which $\mathcal{P}$ vanishes. (I.e., $\mathcal{P}(\mathbf{o}) = 0$ for all $\mathbf{o}$ in O.) Knowledge of this oil space allows a user to efficiently sample preimages for $\mathcal{P}$. This trapdoor can be converted into a post-quantum signature scheme with the Full Domain Hash approach: to sign a message M, the signer produces a preimage $\mathbf{x}$ such that $\mathcal{P}(\mathbf{x}) = H(M)$, where H is a hash function that outputs elements of $\mathbb{F}_q^m$.

This work was supported by CyberSecurity Research Flanders with reference number VR20192203 and the Research Council KU Leuven grant C14/18/067 on Cryptanalysis of post-quantum cryptography. Ward Beullens is funded by a Junior Postdoctoral Fellowship from the Research Foundation - Flanders (FWO), FWO fellowship 1S95620N.

R. AlTawy and A. Hülsing (Eds.): SAC 2021, LNCS 13203, pp. 355–376, 2022.
https://doi.org/10.1007/978-3-030-99277-4_17

Clearly, the security of the scheme relies on the assumption that given $\mathcal{P}$, it is hard to find the oil space $O \subset \mathbb{F}_q^n$ on which $\mathcal{P}$ vanishes. Not surprisingly, if we increase n for fixed $m = \dim(O)$, then finding O becomes more difficult. Initially Patarin proposed to use $n = 2m$, but Kipnis and Shamir showed that in this case O can be found in polynomial time. Their attack runs in time $\tilde{\mathcal{O}}(q^{n-m})$[1], so the attack quickly becomes infeasible if n is sufficiently larger than $2m$. This is why Kipnis *et al.* proposed to use UOV with $n = 3m$. Despite recent progress in key recovery algorithms [1] (which breaks a parameter set with $n = 2.4m$), the $n = 3m$ proposal still seems secure today.

The main drawback of the UOV scheme is that the public keys are large. A public key consists of a list of m multivariate quadratic polynomials in n variables, which requires $\mathcal{O}(mn^2 \log q)$ bits to represent. For example, conservative parameters targeting NIST security level 1 are $m = 53, n = 3m, q = 31$, which results in a key size of 421 KB. Petzoldt *et al.* [10] realized that it is possible to generate a large part of the public key with a PRNG and choose the remaining part such that $\mathcal{P}$ vanishes on a secret space O. This technique allows to reduce the key size from $\mathcal{O}(mn^2 \log q)$ to $O(m^3 \log q)$, which is a significant reduction. For the previous example, this reduces the key size from 421 KB to 48 KB. However, the public key remains large compared to other post-quantum signature schemes.

Contributions. For the UOV trapdoor to work, the dimension of the oil space needs to be at least as large as the number of polynomials m. In this paper, we propose a signing algorithm that uses a UOV map with $o = \dim(O) < m$, which has two immediate benefits:

- By reducing $\dim(O)$, the complexity of key recovery attacks increases, which allows us to choose smaller parameters.
- If $\dim(O)$ is smaller, the constraint that $\mathcal{P}$ vanishes on O becomes weaker, so we can generate a larger part of $\mathcal{P}$ pseudo-randomly with the technique of Petzoldt *et al.* [10]. This reduces the overall key size significantly. We get a key size of $O(mo^2 \log q)$ instead of $O(m^3 \log q)$.

To achieve this, we show how to "whip up" the oil and vinegar: given a UOV map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ that vanishes on some unknown oil space of dimension o, one can construct a larger map $\mathcal{P}^\star : \mathbb{F}_q^{kn} \to \mathbb{F}_q^m$ that vanishes on a space of dimension ko. A simple example of such a map is given by $\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \mathcal{P}(\mathbf{x}_1) + \cdots + \mathcal{P}(\mathbf{x}_k)$, although we will see that this choice of $\mathcal{P}^\star$ will not result in a secure signature scheme. Using this technique, the signature scheme is simple: The public key is a UOV map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ with an oil space of dimension $o < m$. Both the signer and the verifier locally whip up this map to get the larger map $\mathcal{P}^\star$ with an oil space of size $ko \geq m$, which they use as if it was a standard UOV trapdoor.

The case where $k = 1$ (no whipping) and $o = m$ is equivalent to the standard UOV signature scheme, but choosing larger k allows us to reduce o to $\lceil m/k \rceil$, so that we achieve the advantages mentioned earlier.

[1] The $\tilde{O}$-notation ignores polynomial factors.

In this paper, we analyze the security of this construction. We formulate two hard problems, and we show if these problems are indeed hard, then the MAYO scheme is EUF-CMA secure in the random oracle model. Since one of the hardness assumptions is new, this security reduction itself provides little to no evidence for the security of MAYO. However, we hope that by carefully formulating our assumptions, we can help others to understand and cryptanalyze our scheme.

We propose parameter sets aiming for NIST security level I, III, and V. For example, targeting NIST security level I, we propose and implement the parameter set $q = 31, n = 62, m = 60, o = 6$, and $k = 10$. This results in a signature size of 420 bytes, and a public key size of only 803 bytes, which is two orders of magnitude smaller than classic UOV public keys, and even more compact than lattice-based signature schemes such as Falcon [11] and Dilithium [9]. With our implementation, the signing operation takes roughly 1 ms and the verification operation takes 0.5 ms on an intel i5-8400H CPU. Our hope is that the good communication sizes and performance numbers of MAYO will motivate external cryptanalysis of our scheme.

2 Preliminaries

Notation. We denote by $\mathbb{F}_q$ the finite field of q elements. If X is a finite set, we write $x \leftarrow X$ to denote sampling an element from X uniformly at random and assigning the result to x. If A is a (possibly probabilistic) algorithm, we write $y \leftarrow A(x)$ to denote running the algorithm A on input x, and assigning the output to y. We denote the n-by-n identity matrix by $\mathbf{I}_n$. For a square matrix $\mathbf{A} = \{a_{ij}\}_{1 \leq i,j \leq n}$, we denote by $\mathsf{Upper}(\mathbf{A})$ the upper diagonal matrix that is equal to $\mathbf{A}$ up to the addition of an anti-symmetric matrix, i.e., $\mathsf{Upper}(\mathbf{A}) = \{b_{ij}\}_{\leq i,j \leq n}$, where $b_{ij} = a_{ij} + a_{ji}$ if $i \leq j$, $b_{ij} = a_{ij}$ if $i = j$ or $b_{ij} = 0$ otherwise. We say a function $f(\lambda) : \mathbb{N} \to \mathbb{R}$ is negligible if for every $c > 0$, there exits λ_0 such that $|f(\lambda)| < \lambda^{-c}$ for all $\lambda > \lambda_0$.

Multivariate Quadratic Maps. The central object in Multivariate Quadratic cryptography is the multivariate quadratic map. A multivariate quadratic map $\mathcal{P}$ over $\mathbb{F}_q$ with n variables and m components is a sequence $p_1(\mathbf{x}), \cdots, p_m(\mathbf{x})$ of m multivariate quadratic polynomials in n variables $\mathbf{x} = (x_1, \cdots, x_n)$, with coefficients in a finite field $\mathbb{F}_q$. We denote the set of multivariate quadratic maps over $\mathbb{F}_q^n$ with n variables and m components by $\mathsf{MQ}_{n,m,q}$.

To evaluate a map $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$ at a value $\mathbf{a} \in \mathbb{F}_q^n$, we simply evaluate each of its component polynomials in $\mathbf{a}$ to get a vector $\mathbf{b} = (b_1 = p_1(\mathbf{a}), \cdots, b_m = p_m(\mathbf{a}))$ of m output elements. We denote this by $\mathcal{P}(\mathbf{a}) = \mathbf{b}$.

MQ Problem. The main source of computational hardness for multivariate cryptosystems is the Multivariate Quadratic (MQ) problem. Given a multivariate quadratic map $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$, and given a target $\mathbf{t} \in \mathbb{F}_q^m$, the MQ problem asks to find a solution $\mathbf{s}$ such that $\mathcal{P}(\mathbf{s}) = \mathbf{t}$. This problem is NP-hard, and even

though it can be solved in polynomial time if $m \geq n(n+1)/2$ or $n \geq m(m+1)$, it is believed to be exponentially hard on average if $n \sim m$, even for quantum algorithms. Currently, the best algorithms to solve instances of this problem (for cryptographically relevant parameters) are algorithms such as F_4/F_5 or XL that use a Gröbner-basis-like approach [4,6].

Polar Forms. To a homogeneous multivariate quadratic polynomial $p(\mathbf{x})$, we can associate the symmetric bilinear form

$$p'(\mathbf{x}, \mathbf{y}) := p(\mathbf{x} + \mathbf{y}) - p(\mathbf{x}) - p(\mathbf{y}),$$

which is called the *polar form* of $p(\mathbf{x})$. Similarly, we define the polar form of a multivariate quadratic map $\mathcal{P}(\mathbf{x}) = p_1(\mathbf{x}), \cdots, p_m(\mathbf{x})$, to be $\mathcal{P}'(\mathbf{x}, \mathbf{y}) = p'_1(\mathbf{x}, \mathbf{y}), \cdots, p'_m(\mathbf{x}, \mathbf{y})$.

3 The UOV Signature Scheme

As mentioned in the introduction, the Oil and Vinegar signature scheme is based on an elegant multivariate quadratic trapdoor function $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$. This trapdoor function is converted into a signature scheme with the Full Domain Hash approach: The public key is a description of the trapdoor function $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$, the secret key contains the trapdoor information, and a signature on a message M is simply an input $\mathbf{s}$ such that $\mathcal{P}(\mathbf{s}) = \mathcal{H}(M||\mathsf{salt})$, where $\mathcal{H}$ is a cryptographic hash function that outputs elements in the range of $\mathcal{P}$ and where salt is a bit string of length 2λ, chosen at random when the signature is generated. Therefore, to understand the UOV signature scheme, we only need to understand how the UOV trapdoor function works.

3.1 UOV Trapdoor Function

The UOV trapdoor function is a multivariate quadratic map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ that vanishes on a secret linear subspace $O \subset \mathbb{F}_q^n$ of dimension $\dim(O) = m$, i.e.

$$\mathcal{P}(\mathbf{o}) = 0 \quad \text{for all } \mathbf{o} \in O.$$

The trapdoor information is nothing more than a basis for O. To generate the trapdoor function one first picks the subspace O uniformly at random and then one picks $\mathcal{P}$ uniformly at random from the set of multivariate quadratic maps with m components in n variables that vanish on O. Note that on top of the q^m "artificial" zeros in the subspace O, we expect roughly q^{n-m} "natural" zeros that do not lie in O.

Given a target $\mathbf{t} \in \mathbb{F}_q^m$, how do we use this trapdoor to find $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathcal{P}(\mathbf{x}) = \mathbf{t}$? To do this, one picks a vector $\mathbf{v} \in \mathbb{F}_q^n$ and solves the system $\mathcal{P}(\mathbf{v} + \mathbf{o}) = \mathbf{t}$ for a vector $\mathbf{o} \in O$. This can simply be done by solving a linear system for $\mathbf{o}$, because

$$\mathcal{P}(\mathbf{v} + \mathbf{o}) = \underbrace{\mathcal{P}(\mathbf{v})}_{\text{fixed by choice of } \mathbf{v}} + \underbrace{\mathcal{P}(\mathbf{o})}_{=0} + \underbrace{\mathcal{P}'(\mathbf{v}, \mathbf{o})}_{\text{linear function of } \mathbf{o}} = \mathbf{t}.$$

With probability roughly $1 - 1/q$ over the choice of $\mathbf{v}$ the linear map $\mathcal{P}'(\mathbf{v}, \cdot)$ will be non-singular, in which case the linear system $\mathcal{P}(\mathbf{v} + \mathbf{o}) = \mathbf{t}$ has a unique solution. If this is not the case, one can simply pick a new value for $\mathbf{v}$ and try again.

Oil Space Can have Basis of the Form $\left(\mathbf{O}\ \mathbf{I_0}\right)^\top$. In practice, we choose O as the row space of a random matrix of the form $\left(\mathbf{O}\ \mathbf{I}_o\right) \in \mathbb{F}_q^{o \times n}$. Since most o-dimensional subspaces can be represented in this form, this restriction does not affect the security of the scheme much.

Last m Entries of $\mathbf{v}$ Can be Zero. In the original Oil and Vinegar signature scheme the vector $\mathbf{v}$ is not chosen uniformly at random, but the last m entries are fixed to zero. This is slightly more efficient, and it does not affect the output distribution of the signing algorithm. To see why this is the case, notice that adding a vector $\mathbf{o}^\star \in O$ to the choice for $\mathbf{v}$ does not affect the output of the signing algorithm: If $\mathbf{o}$ was the solution to $\mathcal{P}(\mathbf{v} + \mathbf{o}) = \mathbf{t}$, then $\mathbf{o} - \mathbf{o}^\star$ is the solution to $\mathcal{P}(\mathbf{v} + \mathbf{o}^\star + \mathbf{o}') = \mathbf{t}$, so the signing algorithm outputs $\mathbf{v} + \mathbf{o}$ if it started from $\mathbf{v}$, or it outputs $(\mathbf{v} + \mathbf{o}^\star) + (\mathbf{o} - \mathbf{o}^\star)$ if it starts from $\mathbf{v} + \mathbf{o}^\star$. Either way, the output is the same. Therefore, since every $\mathbf{v} \in \mathbb{F}_q^n$ can be written as $\mathbf{v}' + \mathbf{o}$, where the last m entries of $\mathbf{v}'$ are zero, it follows that the last m entries of $\mathbf{v}$ can be fixed at zero without affecting the distribution of the signatures.

4 Key Recovery Attacks Against UOV

A straightforward approach to attack the UOV signature scheme is to completely ignore the existence of the oil subspace and directly try to solve the system $\mathcal{P}(\mathbf{s}) = \mathcal{H}(M||\mathsf{salt})$ to produce a signature for the message M. This can be done with a Gröbner basis-like approach such as XL or F_4/F_5 [4,6]. This is called a direct attack.

More interestingly, the attacker can first try to find the oil space O. After O is found, the attacker can sign any message as if he was a legitimate signer. It was shown by Kipnis and Shamir [8], that O can be found in polynomial time if $n = 2m$, which was the cased for the original oil and vinegar proposal. That is why the current proposals use $n > 2m$, which is known as the Unbalanced Oil and Vinegar (UOV) signature scheme. The conservative recommendation is to use $n = 3m$ or even $n = 4m$, and with these choices there are no known attacks that outperform a direct attack.

In the remainder of this section we summarize the known algorithms for recovering a linear subspace O of dimension o, given a multivariate quadratic map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ that vanishes on this subspace O. Usually, these algorithms are specialized to $o = m$, since this corresponds to the UOV signature use-case. Here, we will generalize the attacks to the case where o is not necessarily equal to m because this is relevant for MAYO. The presentation of the attacks is mostly borrowed from Beullens [1], with slight modifications to generalize to the $o \leq m$ case.

4.1 Reconciliation Attack

The reconciliation attack was developed by Ding *et al.*. as a stepping stone towards the Rainbow Band Separation (RBS) attack against the Rainbow signature scheme [5].

The attack tries to find a number of vectors $\mathbf{o}_1, \mathbf{o}_2, \ldots$ in O, until a complete basis for O is found. To find the first vector $\mathbf{o}_1$ we simply try to find a solution to the system $\mathcal{P}(\mathbf{o}_1) = 0$. By assumption, this system of equations has a o-dimensional linear space of solutions, so if we impose o affine constraints on the entries of $\mathbf{o}_1$, we expect a unique solution $\mathbf{o}_1 \in O$ such that $\mathcal{P}(\mathbf{o}_1) = 0$. This step amounts to finding a solution to a system of m equations in $n - o$ variables, because we can use the o affine constraints to eliminate o variables in the system.

Once the first vector $\mathbf{o}_1 \in O$ is found, it becomes easier to find additional vectors, because the second vector $\mathbf{o}_2$ satisfies $\mathcal{P}(\mathbf{o}_2) = 0$, as well as $\mathcal{P}'(\mathbf{o}_1, \mathbf{o}_2) = 0$, which for fixed $\mathbf{o}_1$ is a set of m linear equations in the entries of $\mathbf{o}_2$. Therefore, after imposing o additional affine constraints, the second step amounts to solving a system of m quadratic equations in $n - m - o$ variables. Compared to the first step, the number of variables is reduced by m, which makes the second step much more efficient. Similarly, finding subsequent vectors $\mathbf{o}_i \in O$ amounts to finding a solution to the system

$$\begin{cases} \mathcal{P}(\mathbf{o}_i) = 0 \\ \mathcal{P}'(\mathbf{o}_1, \mathbf{o}_i) = 0 \\ \ldots \\ \mathcal{P}'(\mathbf{o}_{i-1}, \mathbf{o}_i) = 0 \end{cases},$$

which after imposing o additional affine constraints and eliminating variables amounts to solving a system of m quadratic equations in $n - (i-1)m - o$ variables. If $n < (i-1)m + o$, then we can ignore the quadratic equations and just solve a system of linear equations to find $\mathbf{o}_i$.

The attack does not work as described if $n - o > m$, because in this case the first system $\mathcal{P}(\mathbf{o}_1) = 0$ is underdetermined, and the system has $O(q^{n-o-m})$ solutions, only one of which lies in O. If you start with a solution $\mathbf{o}_1 \notin O$, the subsequent steps will fail to find additional vectors $\mathbf{o}_2, \ldots, \mathbf{o}_o$. In this case one can enumerate all the solutions $\mathcal{P}(\mathbf{o}_1) = 0$, or solve the system

$$\begin{cases} \mathcal{P}(\mathbf{o}_1) = 0 \\ \mathcal{P}(\mathbf{o}_2) = 0 \\ \mathcal{P}'(\mathbf{o}_1, \mathbf{o}_2) = 0 \end{cases},$$

to find $\mathbf{o}_1$ and $\mathbf{o}_2$ simultaneously. In this paper, we will only use UOV maps with $n - o \leq m$, so this more complicated attack is not relevant for us.

If $n - o \leq m$, then the complexity of the attack is dominated by the complexity of finding the first oil vector $\mathbf{o}_1$, which is the complexity of solving a system of m quadratic equations in $n - o$ variables.

4.2 Kipnis-Shamir Attack

Historically, the first attack on the OV signature scheme was given by Kipnis and Shamir [8]. The basic version of this attack works when $n = 2o$, which was the case for the parameter sets initially proposed by Patarin.

Attack if $\mathbf{n = 2o}$. The attack looks at the m components of $\mathcal{P}'(\mathbf{x}, \mathbf{y})$. Each component $p'_i(\mathbf{x}, \mathbf{y}) = p_i(\mathbf{x} + \mathbf{y}) - p_i(\mathbf{x}) - p_i(\mathbf{y})$, defines a matrix M_i such that $p'_i(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top M_i \mathbf{y}$. Kipnis and Shamir observed the following useful property of M_i.

Lemma 1. *For each $i \in \{1, \cdots, m\}$, we have that $M_i O \subset O^\perp$. That is, each M_i sends O into its own orthogonal complement $O^\perp$.*

Proof. For any $\mathbf{o}_1, \mathbf{o}_2 \in O$ we need to prove that $\langle \mathbf{o}_2, M_i \mathbf{o}_1 \rangle = 0$. This follows from the assumption that p_i vanishes on O:

$$\langle \mathbf{o}_2, M_i \mathbf{o}_1 \rangle = \mathbf{o}_2^\top M_i \mathbf{o}_1 = p'_i(\mathbf{o}_1, \mathbf{o}_2) = p_i(\mathbf{o}_1 + \mathbf{o}_2) - p_i(\mathbf{o}_1) - p_i(\mathbf{o}_2) = 0. \quad \square$$

If $n = 2o$, then $\dim(O^\perp) = n - o = o$, so if M_i is nonsingular (which happens with high probability if q is odd), then Lemma 1 turns into an equality $M_i O = O^\perp$. This means that for any pair of invertible M_i, M_j, we have that $M_j^{-1} M_i O = O$, i.e. that O is an invariant subspace of $M_j^{-1} M_i$. It turns out that finding a common invariant subspace of a large number of linear maps can be done in polynomial time, so this gives an efficient algorithm for finding O. For more details we refer to [8].

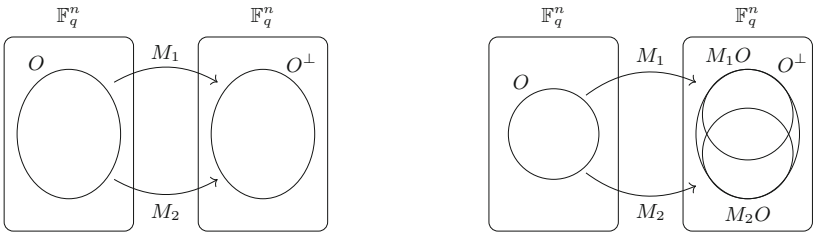

Fig. 1. Behavior of O under M_1 and M_2, in case $n = 2o$ (on the left) and $2o < n < 3o$ (on the right).

Attack if $\mathbf{n > 2o}$. If $n > 2o$, then it is still the case that M_i sends O into $O^\perp$, but because $\dim(O^\perp) = n - o > o$ the equality $M_i O = M_j O$ may no longer hold. Therefore, $M_i^{-1} M_j$ is no longer guaranteed to have O as an invariant subspace and the basic attack fails. However, even though in general $M_i O \neq M_j O$, they still have an unusually large intersection (see Fig. 1): $M_i O$ and $M_j O$ are both subspaces of $O^\perp$, so their intersection has dimension at least $\dim(M_i O) + \dim(M_j O) - \dim(O^\perp) = 3o - n$. Kipnis *et al.* [7] realized that this means that vectors in O are more likely to be eigenvectors of $M_j^{-1} M_i$.

Heuristically, for $\mathbf{x} \in O$, the probability that it gets mapped by M_i to some point in the intersection $M_iO \cap M_jO$ is approximately

$$\frac{|M_iO \cap M_jO|}{|M_iO|} = q^{2o-n}.$$

If this happens, then the probability that M_j^{-1} maps $M_i\mathbf{x}$ back to a multiple of $\mathbf{x}$ is expected to be $(q-1)/|O| \approx q^{1-o}$. Therefore, we can estimate that the probability that a vector in O is an eigenvector of $M_j^{-1}M_i$ is approximately q^{1+o-n}, and the expected number of eigenvectors in O is therefore q^{1+2o-n}.

The same analysis holds when you replace M_i and M_j by arbitrary invertible linear combinations of the M_i. The attacker can repeatedly compute the eigenvectors of $F^{-1}G$, where F and G are random invertible linear combinations of the M_i. After q^{n-2o} attempts he can expect to find a vector in O (he can verify whether a given eigenvector $\mathbf{x}$ is in O by checking that $\mathcal{P}(\mathbf{x}) = 0$). The complexity of the attack is $\tilde{O}(q^{n-2o})$, so the attack runs in polynomial time if $n = 2o$, but quickly becomes infeasible for unbalanced instances of the OV construction. For more details on the attack, we refer to [7].

4.3 Intersection Attack

The intersection attack, introduced by Beullens [1], is a generalisation of the reconciliation attack which uses the ideas behind the Kipnis-Shamir attack. After choosing k matrices $M_1, \ldots, M_k$ as in the Kipnis-Shamir attack, the attacker tries to find a vector $\mathbf{x}$ in the intersection $M_1O \cap \cdots \cap M_kO$. This intersection has dimension at least $ko - (k-1)(n-o)$, so the attacker chooses k such that this is strictly positive. If a vector $\mathbf{x}$ is in this intersection, then $M_i^{-1}\mathbf{x} \in O$ for all $i \in \{1, \ldots, k\}$, which means that $\mathbf{x}$ satisfies the following system of equations:

$$\begin{cases} \mathcal{P}(M_i^{-1}\mathbf{x}) = 0 & \forall i \in \{1, \ldots, k\} \\ \mathcal{P}'(M_i^{-1}\mathbf{x}, M_j^{-1}\mathbf{x}) & \forall i < j \in \{1, \ldots, k\}^2 \end{cases}. \tag{1}$$

The attacker uses a Gröbner-basis-like algorithm to find a solution $\mathbf{x}$ to this system, and recovers k vectors $M_1^{-1}\mathbf{x}, \ldots, M_k^{-1}\mathbf{x}$ in O. Extending these to a basis of O can be done efficiently, as described in Sect. 4.1.

The complexity of the intersection attack is dominated by the complexity of solving a system of $\binom{k+1}{2}m - 2\binom{k}{2}$ linearly independent multivariate quadratic equations (the $\binom{k+1}{2}m$ equations in (1) are linearly dependent) in $n - \dim(M_1O \cap \cdots \cap M_kO) = kn - (2k-1)o$ variables. For more details, we refer to [1].

5 Whipping Oil and Vinegar

In this section we introduce a "whipping" transformation, that turns a multivariate quadratic map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ into a larger map $\mathcal{P}^\star : \mathbb{F}_q^{kn} \to \mathbb{F}_q^m$ for an integer $k > 1$. Our whipping transformation has the property that if $\mathcal{P}(\mathbf{x})$ vanishes on a subspace $O \subset \mathbb{F}_q^n$, then $\mathcal{P}^\star$ vanishes on $O^k \subset \mathbb{F}_q^{kn}$. This allows us to transform a useless UOV map with $o < m$ into a more useful map that vanishes on a space of dimension at least m.

First Attempt. A first attempt is to simply use

$$\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \mathcal{P}(\mathbf{x}_1) + \cdots + \mathcal{P}(\mathbf{x}_k).$$

If $\mathcal{P}$ vanishes on O, then clearly this $\mathcal{P}^\star$ vanishes on O^k. However, it turns out that this $\mathcal{P}^\star$ is not preimage resistant for $k > 1$, so we can not use this construction for our signature scheme. To illustrate the problem, suppose $k \geq 2$ and suppose there exists $\alpha \in \mathbb{F}_q$ such that $\alpha^2 = -1$. Then the attacker can choose $\delta \in \mathbb{F}_q^n$ at random, put $\mathbf{x}_2 = \alpha\mathbf{x}_1 + \delta$, and put $\mathbf{x}_i = 0$ for $i > 2$. Then we have

$$\begin{aligned}\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) &= \mathcal{P}(\mathbf{x}_1) + \mathcal{P}(\alpha\mathbf{x}_1 + \delta) \\ &= \mathcal{P}(\mathbf{x}_1) + \mathcal{P}(\alpha\mathbf{x}_1) + \mathcal{P}(\delta) + \mathcal{P}'(\alpha\mathbf{x}_1, \delta) \\ &= \mathcal{P}(\delta) + \mathcal{P}'(\alpha\mathbf{x}_1, \delta),\end{aligned}$$

where we have used that $\mathcal{P}$ is homogeneous, such that $\mathcal{P}(\alpha\mathbf{x}_1) = -\mathcal{P}(\mathbf{x}_1)$. What remains is linear in $\mathbf{x}_1$, so an attacker can efficiently solve for $\mathbf{x}_1$ such that $\mathcal{P}^\star(\mathbf{x}_1, \alpha\mathbf{x}_1 + \delta, 0, \ldots, 0) = \mathbf{t}$.

Second Attempt. The first attempt resulted in a whipped up map that could be made to collapse into a linear map. To fix this problem, we will add some "emulsifier" maps to the mix.[2] Concretely, for the second attempt we choose k invertible linear m-by-m matrices $\mathbf{E}_1, \ldots, \mathbf{E}_k$ at random and set

$$\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \mathbf{E}_1\mathcal{P}(\mathbf{x}_1) + \cdots + \mathbf{E}_k\mathcal{P}(\mathbf{x}_k).$$

This blocks attacks of the type that broke our first attempt: Suppose the attacker sets $\mathbf{x}_i = \alpha_i\mathbf{x}_1 + \delta_i$, for $i > 1$ and for some $\alpha_i \in \mathbb{F}_q$ and $\delta_i \in \mathbb{F}_q^n$, then the quadratic part of $\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ becomes

$$\left(\mathbf{E}_1 + \sum_{i=2}^{k} \alpha_i^2 \mathbf{E}_i\right) \mathcal{P}(\mathbf{x}_1).$$

If the $\mathbf{E}_i$ are chosen at random, then for each choice of α_i, the probability that the quadratic terms vanish is q^{-m^2}, so a union bound says that the probability that there exist α_i such that the quadratic part vanishes is at most q^{k-1-m^2}, which can be made negligibly small by choosing the parameters appropriately. However, the attacker can still take advantage of α_i such that $\mathbf{E}_1 + \sum_{i=2}^{k} \alpha_i^2 \mathbf{E}_i$ has low rank. Therefore, we choose the $\mathbf{E}_i$ from a set of q^m matrices such that any non-zero linear combination of these matrices has full rank. We use the set of matrices that correspond to multiplication by elements of $\mathbb{F}_{q^m}$. In the following, we fix an embedding of $\mathbb{F}_{q^m}$ in the algebra of m-by-m matrices over $\mathbb{F}_q$, and

[2] An emulsifier is a chemical that stabilizes an emulsion. An example is Lecithin, which is found in egg yolks, and which can stabilize a foam of oil droplets in an oil and vinegar mixture to form mayonnaise.

with a mild abuse of notation, we will identify the elements of $\mathbb{F}_{q^m}$ with the corresponding matrices. With this choice of "emulsifier maps", the probability that there exists a linear combination $\mathbf{E}_1 + \sum_{i=2}^{k} \alpha_i^2 \mathbf{E}_i$ with rank lower than n (i.e. rank 0) is at most q^{k-1-m}, which can still be made negligible.[3]

However, there is still a different issue. Since $\mathcal{P}^\star$ is the sum of k functions with independent inputs the problem of finding a preimage for $\mathcal{P}^\star$ reduces to a k-SUM problem. The attacker constructs k lists of evaluations of $\mathbf{E}_i(\mathcal{P}(\mathbf{x}))$ respectively, and searches for one value in each list such that their sum is $\mathbf{t}$. This can be done in time $O(q^{m/\lfloor log_2(k) \rfloor})$ with Wagner's k-tree algorithm [14]. For moderately large values of k (e.g. $k = 8$) this attack will be more efficient than the other known attacks against our signature scheme, so it is worthwhile to choose a different $\mathcal{P}^\star$ that is not susceptible to this attack.

Final Construction. To avoid the k-tree attacks, we finally propose to use the following construction: Let q be odd, choose invertible linear matrices $\mathbf{E}_{i,j}$ for all (i, j) with $1 \leq i \leq j \leq n$ (still representing multiplication by an element of $\mathbb{F}_{q^m}$), and let

$$\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \sum_{1 \leq i \leq j \leq n} \mathbf{E}_{i,j}(\mathcal{P}(\mathbf{x}_i + \mathbf{x}_j)).$$

Remark 2. Note that in characteristic 2, the $\mathbf{E}_{ii}\mathcal{P}(\mathbf{x}_i + \mathbf{x}_i)$ terms vanish, so it would be slightly more natural to consider the maps $\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k) = \sum_i \mathbf{E}_{ii}\mathcal{P}(\mathbf{x}_i) + \sum_{i<j} \mathbf{E}_{ij}\mathcal{P}(\mathbf{x}_i + \mathbf{x}_j)$. Both definitions are equivalent for odd q, so to keep the notation (and the implementation of our scheme) as simple as possible, we have chosen to use the simpler definition, and to let q be odd.

The probability that there exist α_i such that the quadratic part of $\mathcal{P}^\star(\mathbf{x}_1, \alpha_2\mathbf{x}_2 + \delta_2, \ldots, \alpha_k\mathbf{x}_1 + \delta_1)$ is still bounded by q^{k-1-m}. Moreover, the cross-terms in $\mathbf{E}_{i,j}\mathcal{P}(\mathbf{x}_i + \mathbf{x}_j)$ prevent the list-sum attack, because in general $\mathcal{P}^\star(0, \ldots, \mathbf{x}_i, \ldots, 0) + \mathcal{P}^\star(0, \ldots, \mathbf{x}_j, \ldots, 0) \neq \mathcal{P}^\star(0, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_j, \ldots, 0)$.

6 Mayo Signatures

In this section we introduce our new signature scheme that uses UOV maps with $o < m$. Recall that in the $o = m$ case, the signature generation algorithms proceeds by picking a random salt of length 2λ and a random vector $\mathbf{v} \in \mathbb{F}_q^n$, and solving for $\mathbf{o} \in O$ such that $\mathcal{P}(\mathbf{v} + \mathbf{o}) = \mathsf{Hash}(M||\mathsf{salt})$, which is a linear system of equations. If $o < m$ the same strategy fails because the linear system has m equations, but only $o < m$ degrees of freedom, such that with large probability the system will not have any solutions. To solve this problem, we fix some k such that $ko \geq m$ and we let the signer whip up $\mathcal{P}(\mathbf{x})$ into a larger map $\mathcal{P}^\star(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ with the method from the previous section (with random

[3] For odd q we can get a slightly better bound of $\left(\frac{q+1}{2}\right)^{k-1} q^{-m}$, because each α_i^2 can only take $(q + 1)/2$ distinct values.

emulsifier maps $\{\mathbf{E}_{ij}\}_{1\leq i\leq j\leq k}$ obtained by hashing $M||\mathsf{salt}$). Now the signer can choose $(\mathbf{v}_1,\ldots,\mathbf{v}_k)\in\mathbb{F}_q^{kn}$, and solve for $(\mathbf{o}_1,\ldots,\mathbf{o}_k)\in O^k$ such that $\mathcal{P}(\mathbf{v}_1+\mathbf{o}_1,\ldots,\mathbf{v}_k+\mathbf{o}_k)=\mathbf{t}$. This amounts to solving a system of m linear equations with $ko\geq m$ degrees of freedom, so solutions can be found with large probability. The signature consists of the salt, and the preimage $\{\mathbf{s}_i=\mathbf{v}_i+\mathbf{o}_i\}_{i\in[k]}$. Note that, as in the original UOV signature algorithm, we can let the last o entries of the $\mathbf{v}_i$ be zero to speed up the signing algorithm without affecting its output distribution.

To verify a signature, the verifier simply hashes $M||\mathsf{salt}$ to obtain $\{\mathbf{E}_{ij}\}_{1\leq i\leq j\leq k}$ and $\mathbf{t}$, and accepts the signature if and only if $\mathcal{P}^\star(\mathbf{s}_i)=\mathbf{t}$.

To generate a key-pair, a user first chooses a random oilspace by sampling a uniformly random o-by-$(n-o)$ matrix $\mathbf{O}$, and letting O be the rowspace of $(\mathbf{O}\,\mathbf{I}_o)$, where $\mathbf{I}_o$ is the identity matrix of size o. Then the user generates a random multivariate quadratic map $\mathcal{P}(\mathbf{x})$ that vanishes on O. Recall that every multivariate quadratic polynomial $p_i(\mathbf{x})$ of the public key can be represented with an upper triangular matrix $\mathbf{P}_i$ such that

$$p_i(\mathbf{x})=\mathbf{x}^\top\mathbf{P}_i\mathbf{x}=\mathbf{x}^\top\begin{pmatrix}\mathbf{P}_i^{(i)} & \mathbf{P}_i^{(2)}\\ 0 & \mathbf{P}_i^{(3)}\end{pmatrix}\mathbf{x},$$

where $\mathbf{P}_i^{(1)}$ and $\mathbf{P}_i^{(3)}$ are square upper triangular matrices of size $n-o$ and o respectively, and where $\mathbf{P}_i^{(2)}$ is rectangular of size $(n-o)$-by-o. To reduce the size of the public key, we choose the matrices $\mathbf{P}_i^{(i)}$ and $\mathbf{P}_i^{(2)}$ pseudo-randomly from a random seed value $\mathsf{seed}\in\{0,1\}^\lambda$. Then we solve for $\mathbf{P}_i^{(3)}$ such that p_i vanishes on O. The polynomial $p_i(\mathbf{x})$ vanishes on O if

$$(\mathbf{O}\;\mathbf{I}_o)\begin{pmatrix}\mathbf{P}_i^{(i)} & \mathbf{P}_i^{(2)}\\ 0 & \mathbf{P}_i^{(3)}\end{pmatrix}(\mathbf{O}\;\mathbf{I}_o)^\top=\mathbf{O}\mathbf{P}_i^{(1)}\mathbf{O}^\top+\mathbf{O}\mathbf{P}_i^{(2)}+\mathbf{P}_i^{(3)}=0,$$

so it suffices to set $\mathbf{P}_i^{(3)}$ to be $\mathsf{Upper}(-\mathbf{O}\mathbf{P}_i^{(1)}\mathbf{O}^\top-\mathbf{O}\mathbf{P}_i^{(2)})$. Note that taking Upper does not influence the quadratic polynomial represented by $\mathbf{P}_i$.

The key generation, signing and verification algorithms are described in more detail in Fig. 2.

The following lemma says the probability that the signing algorithm needs to restart is small if $ok\geq m$. The proof is not particularly interesting, so in the interest of space we put it in Appendix A.

Lemma 3. *Let $O,\mathcal{P},\{\mathbf{E}_{ij}\}$, and $\{\mathbf{v}_i\}_{1\in[k]}$ in $\mathbb{F}_q^{n-m}\times\{0\}^m$ be chosen at random as during the key-generation and signing algorithms of the MAYO signature scheme with parameters n,m,o,k,q. Then as a function of $\{\mathbf{o}_i\}_{i\in[k]}\in O$ the affine map*

$$\mathcal{P}^\star(\mathbf{v}+\mathbf{o})=\sum_{ij}\mathbf{E}_{ij}\mathcal{P}(\mathbf{v}_i+\mathbf{v}_j+\mathbf{o}_i+\mathbf{o}_k)$$

has full rank except with probability bounded by $\frac{1}{q^m-1}+\frac{q^{k-(n-o)}}{q-1}+\frac{q^{m-ko}}{q-1}$.

KeyGen():

1: $\mathbf{O} \leftarrow \mathbb{F}_q^{o \times (n-o)}$
2: $\mathsf{seed} \leftarrow \{0,1\}^{\lambda}$
3: **for** i from 1 to m **do**
4: $\quad \mathbf{P}_i^{(1)} \leftarrow \mathsf{Expand}(\mathsf{seed}||\mathsf{P1}||i)$ ▷ Upper triangular $(n-o)$-by-$(n-o)$ matrix.
5: $\quad \mathbf{P}_i^{(2)} \leftarrow \mathsf{Expand}(\mathsf{seed}||\mathsf{P2}||i)$ ▷ o-by-$(n-o)$ matrix.
6: $\quad \mathbf{P}_i^{(3)} \leftarrow \mathsf{Upper}(-\mathbf{O}\mathbf{P}_i^{(1)}\mathbf{O}^{\top} - \mathbf{O}\mathbf{P}_i^{(2)})$
7: **return** $(\mathsf{pk}, \mathsf{sk}) = ((\mathsf{seed}, \{\mathbf{P}_i^{(3)}\}_{i \in \{i,\ldots,m\}}), (\mathsf{seed}, \mathbf{O}))$.

Sign(M, sk)**:**

1: $(\mathsf{seed}, \mathbf{O}) \leftarrow \mathsf{sk}$
2: $\mathsf{salt} \leftarrow \{0,1\}^{2\lambda}$
3: $(\{\mathbf{E}_{ij}\}_{1 \le i \le j \le k}, \mathbf{t}) \leftarrow \mathsf{Hash}(M||\mathsf{salt})$
4: $\mathcal{P}^*(\mathbf{x}_1, \ldots \mathbf{x}_k) \leftarrow \sum_{1 \le i \le j \le k} \mathbf{E}_{ij}\mathcal{P}(\mathbf{x}_i + \mathbf{x}_j)$
5: $\mathbf{v}_i \leftarrow \mathbb{F}_q^{n-m} \times \{0\}^m$
6: If $\mathcal{P}^*(\mathbf{v}_1 + \mathbf{o}_1, \ldots, \mathbf{v}_k + \mathbf{o}_k)$ does not have full rank, return to step 2.
7: Solve $\mathcal{P}^*(\mathbf{v}_1 + \mathbf{o}_1, \ldots, \mathbf{v}_k + \mathbf{o}_k) = \mathbf{t}$ for $\mathbf{o}_1, \ldots, \mathbf{o}_k \in \mathrm{RowSpace}((\mathbf{O}\,\mathbf{I}_o))$.
8: **return** $\sigma = (\mathsf{salt}, \{\mathbf{x}_i = \mathbf{v}_i + \mathbf{o}_i\}_{i \in [k]})$

Verify(M, pk, σ)**:**

1: $(\mathsf{salt}, \{\mathbf{x}_i\}_{i \in [k]}) \leftarrow \sigma$
2: $(\{\mathbf{E}_{ij}\}_{1 \le i \le j \le k}, \mathbf{t}) \leftarrow \mathsf{Hash}(M||\mathsf{salt})$
3: $\mathbf{t}' \leftarrow \sum_{1 \le i \le j \le k} \mathbf{E}_{ij}\mathcal{P}(\mathbf{x}_i + \mathbf{x}_j)$
4: **return** accept if $\mathbf{t} = \mathbf{t}'$ and reject otherwise.

Fig. 2. The key generation, signing, and verification algorithms of the MAYO signature scheme.

7 Security Analysis

Traditional MQ signature algorithms usually rely on ad-hoc assumptions, which makes it impossible to prove security reductions from well-established hardness assumptions.[4] The MAYO signature scheme is no exception. However, we will still formally define two assumptions based on which our scheme can be proven to be secure. Since one of the assumptions is new, this security reduction itself does not provide any kind of guarantee for the security of the scheme. Still, we hope the security reduction is valuable for cryptanalysts to understand what is necessary to attack our scheme. Most notably, we prove that if ko is sufficiently larger than m, each signature only leaks a negligible amount of information about the secret key.

Our first hardness assumption says that it is hard to distinguish a random multivariate quadratic map that vanishes on a random linear subspace from a uniformly random quadratic map.

Definition 4 (UOV problem). *For* $\mathbf{O} \in \mathbb{F}_q^{o \times (n-o)}$*, we let* $\mathsf{MQ}_{n,m,q}(\mathbf{O})$ *denote the set of* $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$ *that vanish on the rowspace of* $(\mathbf{O}\ \mathbf{I}_o)$*. The UOV problem*

[4] Signature schemes such as MQDSS [3,12] and MUDFISH [2] that do not make use of trapdoors are an exception because they enjoy security reductions from the one-wayness of a system of uniformly random multivariate quadratic equations.

asks to distinguish a random multivariate quadratic map $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$*, from a random multivariate quadratic map in* $\mathsf{MQ}_{n,m,q}(\mathbf{O})$ *for a random* $\mathbf{O} \in \mathbb{F}_q^{o\times(n-o)}$.

Let $\mathcal{A}$ *be a UOV distinguisher algorithm. We say the distinguishing advantage of* $\mathcal{A}$ *is*

$$\mathsf{Adv}^{\mathsf{UOV}}_{n,m,o,q}(\mathcal{A}) = \Big|\Pr\left[\mathcal{A}(\mathcal{P}) = 1 \middle| \mathcal{P} \leftarrow \mathsf{MQ}_{m,n,q}\right] - \Pr\left[\mathcal{A}(\mathcal{P}) = 1 \middle| \begin{array}{c} \mathbf{O} \leftarrow \mathbb{F}_q^{o\times(n-o)} \\ \mathcal{P} \leftarrow \mathsf{MQ}_{n,m,q}(\mathbf{O}) \end{array}\right]\Big|.$$

The UOV problem has been studied since the invention of the UOV signature scheme in 1997 and seems relatively well understood. In contrast, our second hardness assumption is tailored to the MAYO signature scheme and is therefore a new assumption. This assumption says that picking a random multivariate quadratic map $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$, and whipping it up to a larger map $\mathcal{P}^\star \in \mathsf{MQ}_{kn,m,q}$ results in a preimage resistant function on average.

Definition 5 (Whipped MQ problem). *Given random* $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$, $\{\mathbf{E}_{ij}\}_{1\le i\le j\le k} \in \mathbb{F}_{q^m}$ *and* $\mathbf{t} \in \mathbb{F}_q^m$*, the whipped MQ problem asks to compute* $\mathbf{s}_1, \ldots, \mathbf{s}_k$*, such that* $\sum_{i,j} \mathbf{E}_{ij}\mathcal{P}(\mathbf{s}_i + \mathbf{s}_j) = \mathbf{t}$.

Let $\mathcal{A}$ *be an adversary. We say that the advantage of* $\mathcal{A}$ *against the whipped MQ problem is*

$$\mathsf{Adv}^{\mathsf{WMQ}}_{n,m,k,q}(\mathcal{A}) = \Pr\left[\sum_{i,j} \mathbf{E}_{ij}\mathcal{P}(\mathbf{s}_i + \mathbf{s}_j) = \mathbf{t} \middle| \begin{array}{c} \mathcal{P} \leftarrow \mathsf{MQ}_{n,m,q} \\ \{\mathbf{E}_{ij}\}_{1\le i\le j\le k} \leftarrow \mathbb{F}_{q^m} \\ \mathbf{t} \leftarrow \mathbb{F}_q^m \\ (\mathbf{s}_1, \ldots, \mathbf{s}_k) \leftarrow \mathcal{A}(\mathcal{P}, \{\mathbf{E}_{ij}\}_{i,j}, \mathbf{t}) \end{array}\right].$$

Finally, we state the standard EUF-CMA and EUF-KOA security definition for digital signature algorithms in the random oracle model.

Definition 6 (EUF-CMA/EUF-KOA security). *Let* $\mathcal{O}$ *be a random oracle, and let* $\mathcal{A}$ *be an adversary. We say the advantage of* $\mathcal{A}$ *gainst the* EUF-CMA *game of a signature scheme* $S = (KeyGen, Sign^{\mathcal{O}}, Verify^{\mathcal{O}})$ *in the random oracle model is*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{S}(\mathcal{A}) = \Pr\left[\begin{array}{c} Verify^{\mathcal{O}}(\mathsf{pk}, m, \sigma) = 1, \\ and\ Sign^{\mathcal{O}}(\mathsf{sk}, \cdot)\ was \\ not\ queried\ on\ input\ m \end{array} \middle| \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow KeyGen() \\ (m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}, Sign^{\mathcal{O}}(\mathsf{sk}, \cdot)}(\mathsf{pk}) \end{array}\right].$$

The EUF-KOA *advantage* $\mathsf{Adv}^{\mathsf{EUF\text{-}KOA}}_{S}(\mathcal{A})$ *is defined in the same way, except that* $\mathcal{A}$ *does not have access to the signing oracle* $Sign^{\mathcal{O}}(\mathsf{sk}, \cdot)$.

With these definitions out of the way we can formulate our security theorem.

Theorem 7. *Let* $\mathcal{A}$ *be an* EUF-CMA *adversary that runs in time* T *against the MAYO signature in the random oracle model with parameters* n, m, o, k, q,

and which makes Q_s signing queries and Q_h queries to the random oracle. Let $\mathsf{B} = \frac{1}{q^m-1} + \frac{q^{k-(n-o)}}{q-1} + \frac{q^{m-ko}}{q-1}$ *be the bound on the restarting probability from Lemma 3 and suppose $Q_s\mathsf{B} < 1$, then there exist adversaries $\mathcal{A}_{\mathsf{UOV}}$ and $\mathcal{A}_{\mathsf{WMQ}}$ against the $UOV_{n,m,o,q}$ and $WMQ_{n,m,k,q}$ assumptions respectively, that run in time $T + (Q_s + Q_h + 1) \cdot poly(n, m, k, q)$ such that*

$$\mathsf{Adv}^{\mathsf{EUF-CMA}}_{n,m,o,k,q}(\mathcal{A}) \leq \left(\mathsf{Adv}^{\mathsf{UOV}}_{n,m,o,q}(\mathcal{B}) + Q_h\mathsf{Adv}^{\mathsf{WMQ}}_{n,m,k,q}(\mathcal{B}') + q^{-m}\right)(1 - Q_s\mathsf{B})^{-1} + (Q_h + Q_s)Q_s 2^{-2\lambda}.$$

We prove the theorem with two lemmas. The first lemma reduces the EUF-CMA security of the MAYO signature scheme to its EUF-KOA, by showing that we can simulate a signing oracle if ko is sufficiently larger than m. The second lemma then finishes the proof by giving a reduction from the UOV and WMQ problems to the EUF-KOA security game. The reduction from the WMQ problem loses a factor Q_h in advantage, because the reduction programs the random oracle to output the WMQ instance for one of the Q_h random oracle queries, and succeeds only if the adversary forges a signature for that particular query. The proofs of Lemmas 8 and 9 can be found in Appendices B and C respectively.

Lemma 8. *If there exists an adversary $\mathcal{A}$, that runs in time T against the* EUF-CMA *security of the MAYO signature in the random oracle model with parameters n, m, o, k, q, with $k < (n-o)$, and which makes Q_h queries to the random oracle and Q_s queries to the signing oracle. Let* $\mathsf{B} = \frac{1}{q^m-1} + \frac{q^{k-(n-o)}}{q-1} + \frac{q^{m-ko}}{q-1}$ *be the bound on the restarting probability from Lemma 3 and suppose $Q_s\mathsf{B} < 1$, then there exists an adversary $\mathcal{B}$ against the* EUF-KOA *security of the MAYO signature scheme, that runs in time $T + O((Q_h + Q_s)poly(n, m, k, q))$ such that*

$$\mathsf{Adv}^{\mathsf{EUF-CMA}}_{n,m,o,k,q}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}KOA}}_{n,m,o,q}(\mathcal{B})\,(1 - Q_s\mathsf{B})^{-1} + (Q_h + Q_s)Q_s 2^{-2\lambda}.$$

Lemma 9. *Let $\mathcal{A}$ be an* EUF-KOA *adversary that runs in time T against the MAYO signature in the random oracle model with parameters n, m, o, k, q, and which makes Q_h queries to the random oracle. Then there exists an adversary $\mathcal{B}$ against the $UOV_{n,m,o,q}$ problem, and an adversary $\mathcal{B}'$ against the $WMQ_{n,m,k,q}$ problem, that run in time $T + O((1 + Q_h)poly(n, m, k, q))$ such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}KOA}}_{n,m,o,k,q}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UOV}}_{n,m,o,q}(\mathcal{B}) + (1 + Q_h)\mathsf{Adv}^{\mathsf{WMQ}}_{n,m,k,q}(\mathcal{B}') + q^{-m}.$$

8 Parameter Selection and Implementation

In this section, we choose some parameter sets for the MAYO signature scheme. A parameter set consists of five values n, m, o, k, and q (as well as the length of the salt, which we choose to be 256, 384 or 512 bits long for NIST security levels I, III, and V respectively.) The only requirement for the correctness of the signature scheme is that $ko \geq m$ because otherwise, the signing algorithm

will fail with high probability. For security, we need to choose n, m, o, k and q such that the UOV and WMQ problems are hard. The best known attacks against the UOV assumption are summarized in Sect. 3. Since we are not aware of attacks that exploit the whipping structure, we estimate that the hardness of the WMQ problem is the same as the hardness of breaking the preimage resistance of a uniformly random multivariate quadratic map $\mathcal{P} \in \mathsf{MQ}_{kn,m,q}$. These systems are very underdetermined, so we can use the technique of Thomae and Wolf [13] to reduce the problem of finding a solution to a system in $\mathsf{MQ}_{kn,m,q}$, to a system in $\mathsf{MQ}_{n',m',q}$, where $n' = m' = \lceil m + 1 - \frac{nk}{m} \rceil$. To achieve NISTPQC security levels I, III, or V we choose parameters such that finding such a solution with the Hybrid XL algorithm, or breaking the UOV assumption costs at least $2^{143}, 2^{207}$, or 2^{272} bit operations respectively. The fact that all known attacks require frequently accessing large amounts of memory provides a comfortable security margin. Table 1 contains the proposed parameter sets. Estimates of the bit complexity of known attacks against these parameter sets are given in Table 2.

Our security reduction has a factor Q_h advantage loss for the reduction from the WMQ problem, where Q_h is the number of random oracle queries that the adversary is allowed to make. Therefore, if one wanted the reduction to guarantee l bits of security, we would have to pick parameters such that the WMQ problem has $2l$ bits of hardness. We choose not to do this because it would come at a significant cost in performance and communication size, and we are not aware of any attacks that exploit the looseness in the reduction. E.g., for our parameters, there do not appear to exist multi-target attacks on the WMQ problem that meaningfully outperform single-target attacks. (This is also the case for the standard MQ problem.)

Information-theoretically, UOV signatures (and variants such as Rainbow) leak information about the secret key. Although it seems hard to exploit this leakage in an attack, one might want to stop this leakage altogether. For the UOV scheme, it would be possible to stop the leakage by choosing $o > m$, but this would come at a very significant cost in terms of performance. For the MAYO signatures, it is much cheaper to prevent the leakage, because we only need $ko > m$. Table 1 proposes two parameter sets per NIST security level: a first parameter set that does not attempt to prevent leakage, and a second parameter set that satisfies $\mathsf{B} \leq 2^{-65}$, such that Lemma 8 gives a tight reduction from EUF-KOA security to EUF-CMA security for adversaries that are allowed to make up to 2^{64} signature queries. Figure 3 shows the signature size and public key size of a variety of MAYO parameter sets (with and without leaky signatures), compared to the key and signature sizes of the three finalist signature schemes in the NISTPQC process. We see that by choosing the parameters, we can make a trade-off between signature size and public key size. We also see that the cost of making the signatures statistically close to random is small.

Table 1. Parameter sets for the MAYO signature scheme.

SL	No leakage	Parameters					|pk| (Bytes)	|sig| (Bytes)
		n	m	o	k	q		
I	✗	74	76	6	13	7	614	392
	✓	76	78	7	15	7	835	459
III	✗	111	117	6	20	7	937	880
	✓	112	117	7	20	7	1244	888
V	✗	148	152	8	20	7	2068	1174
	✓	149	157	7	26	7	1664	1516

Table 2. Estimated complexities (log_2 of number of bit operations) of known attacks against MAYO parameter sets.

SL	No leakage	Parameters	Direct	KS	Recon.	Inters.
		$n\ m\ o\ k\ q$				
I	✗	74, 76, 6, 13, 7	145	190	143	245
	✓	76, 78, 7, 15, 7	145	190	144	245
III	✗	111, 117, 6, 20, 7	210	296	208	374
	✓	112, 117, 7, 20, 7	209	292	208	370
V	✗	148, 152, 8, 20, 7	273	390	272	486
	✓	149, 157, 7, 26, 7	273	398	272	498

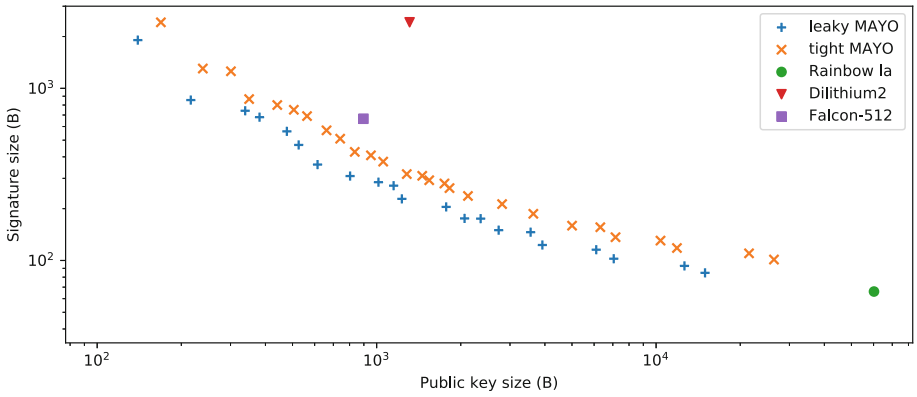

Fig. 3. A comparison of the key and signature sizes of the MAYO signature scheme with various parameter sets, and the key and signature sizes of the NISTPQC signature finalists.

Implementation. We made a C implementation with some preliminary AVX2 optimizations of MAYO for the parameter set $(n = 62, m = 60, o = 6, k = 10, q = 31)$, which aims for NISTPQC security level I. We instantiate the H and Expand random oracles with the SHAKE128 extendable output function. With these choices, the public key and signatures have a size of 803 Bytes and 420 Bytes respectively. On an Intel i5-8400H CPU at 2.5 GHz, a signing operation takes 2.50 million cycles, and a verification operation takes 1.3 million cycles (i.e., 1 ms or 0.5 ms respectively). A large fraction of the time is spent expanding the public seed with Expand, therefore, if one can spare 137 KB to store the expanded seed the signing and verification time can be reduced by 30% and 40%, to 1.7 million cycles and 820 thousand cycles respectively (i.e., 0.7 ms or 0.3 ms). We leave a more optimized constant-time implementation of MAYO for future work.

A Proof of Lemma 3

Before we prove the lemma, we recall the following result, which is useful to prove that certain random matrices are of full rank with high probability. In particular the result applies to uniformly random matrices, and uniformly random symmetric matrices.

Lemma 10. *Let $\mathcal{M}$ be a distribution of matrices in $\mathbb{F}_q^{n\times m}$ with $n \geq m$, such that for all $\mathbf{x} \in \mathbb{F}_q^m \setminus \{0\}$, we have*

$$\Pr_{\mathbf{M}\leftarrow\mathcal{M}}[\mathbf{M}\mathbf{x} = 0] = q^{-n},$$

then the probability that $\mathbf{M} \leftarrow \mathcal{M}$ does not have full rank is bounded by $\frac{q^{m-n}}{q-1}$.

Proof. From the assumption, it follows that the average number of non-zero kernel vectors is $(q^m - 1)q^{-n}$. Since every matrix which does not have full rank has at least $q - 1$ non-zero kernel vectors, it follows that

$$\Pr_{\mathbf{M}\leftarrow\mathcal{M}}[\mathrm{rank}(\mathbf{M}) < m](q-1) \leq (q^m - 1)q^{-n} < q^{m-n}. \qquad \square$$

A.1 Proof of Lemma 3

Proof. First of all, we show that if $\mathbf{v}_1, \cdots, \mathbf{v}_k \in \mathbb{F}_q^{n-o} \times \{0\}^o$ are linearly independent, then the linear maps $\mathcal{P}'(\mathbf{v}_1, \cdot), \ldots, \mathcal{P}'(\mathbf{v}_k, \cdot)$ from O to $\mathbb{F}_q^m$ are all independent and uniformly distributed. To see this, it suffices to show that for a basis $\mathbf{y}_1, \cdots \mathbf{y}_o$ of O, the matrices $\{p'_i(\mathbf{v}_a, \mathbf{y}_b)\}_{a\in[k],b\in[o]}$ are independent and uniformly random for all $i \in [m]$. If we choose the basis where $\mathbf{y}_b$ is the b-th row of $\begin{pmatrix}\mathbf{O} & \mathbf{I}_o\end{pmatrix}$, then a calculation shows that these matrices are

$$\mathbf{V}\left((\mathbf{P}_i^{(1)} + \mathbf{P}_i^{(1)\top})\mathbf{O}^\top + \mathbf{P}_i^{(2)}\right),$$

where the rows of $\mathbf{V} \in \mathbb{F}_q^{k\times(n-o)}$ consists of the first $n - o$ entries of the $\mathbf{v}_i$. Therefore, if the $\mathbf{v}_i$ are linearly independent, then $\mathbf{V}$ has full rank, and if

$k < (n-o)$, then it follows that these matrices are uniformly random and independent because the $\mathbf{P}_i^{(2)}$ matrices are chosen uniformly at random during the key generation algorithm.

In particular, if $\mathbf{M}_1, \ldots, \mathbf{M}_k \in \mathbb{F}_q^{n \times o}$ are the matrix representations of $\mathcal{P}'(\mathbf{v}_i, \cdot)$ (i.e. the matrices such that for all $i \in [k]$, we have $\mathcal{P}'(\mathbf{v}_i, \sum_i u_i \mathbf{y}_i) = \mathbf{M}_i \mathbf{u}$). Then we have shown that if the $\mathbf{v}_i$ are linearly independent, then the $\mathbf{M}_i$ are independent and uniformly random matrices.

As a warm-up, let us now look at the case $k = 1$ first. In this case the linear part of $\mathcal{P}^\star(\mathbf{v}+\mathbf{o})$ is $\mathcal{P}^{\star\prime}(\mathbf{v}, \mathbf{o}) = 4\mathbf{E}_{11}\mathcal{P}'(\mathbf{v}, \mathbf{o})$. This has the matrix representation $\mathbf{E}_{11}\mathbf{M}_1$, where if $\mathbf{v} \neq 0$, the matrix $\mathbf{M}_1$ is uniformly random. Therefore, we see that the signing algorithm has to restart with probability bounded by

$$q^{-m} + q^{o-n} + \frac{q^{m-o}}{q-1}$$

because either $\mathbf{E}_{11} = 0$ or $\mathbf{v} = 0$, which happens with probability bounded by $q^{-m} + q^{o-n}$, and in which case $\mathbf{E}_{11}\mathcal{P}(\mathbf{v}+\mathbf{o})$ is exactly zero, so it definitely is not full rank, or otherwise the linear part of $\mathbf{E}_{11}\mathcal{P}(\mathbf{v}+\mathbf{o})$ is a uniformly random linear map from O to $\mathbb{F}_q^m$, so it fails to have full rank with probability bounded by $\frac{q^{m-o}}{q-1}$ (Lemma 10).

In general, the linear part of $\mathcal{P}^\star(\mathbf{v}+\mathbf{o})$ is equal to

$$\mathcal{P}'^\star(\mathbf{v}, \mathbf{o}) = \sum_{ij} \mathbf{E}_{ij} \mathcal{P}'(\mathbf{v}_i + \mathbf{v}_j, \mathbf{o}_i + \mathbf{o}_j) \tag{2}$$

$$= \sum_{ij} \mathbf{E}_{ij} \left(\mathcal{P}'(\mathbf{v}_i, \mathbf{o}_i) + \mathcal{P}'(\mathbf{v}_i, \mathbf{o}_j) + \mathcal{P}'(\mathbf{v}_j, \mathbf{o}_i) + \mathcal{P}'(\mathbf{v}_j, \mathbf{o}_j)\right) \tag{3}$$

Let $\mathbf{M}_1, \ldots, \mathbf{M}_k$ be the matrix representations of $\mathcal{P}'(\mathbf{v}_i, \cdot)$, then the matrix representation of $\mathcal{P}^{\star\prime}(\mathbf{v}, \cdot)$ is $\begin{pmatrix}\mathbf{M}'_1 & \ldots & \mathbf{M}'_k\end{pmatrix} \in \mathbb{F}_q^{m \times ko}$, where

$$\begin{pmatrix}\mathbf{M}'_1 \\ \vdots \\ \mathbf{M}'_k\end{pmatrix} = \mathbf{E}\begin{pmatrix}\mathbf{M}'_1 \\ \vdots \\ \mathbf{M}'_k\end{pmatrix} = \begin{pmatrix}\mathbf{D}_1 & \mathbf{E}_{12} & \ldots & \mathbf{E}_{1k} \\ \mathbf{E}_{12} & \mathbf{D}_2 & \ldots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{E}_{1k} & \ldots & \ldots & \mathbf{D}_k\end{pmatrix}\begin{pmatrix}\mathbf{M}'_1 \\ \vdots \\ \mathbf{M}'_k\end{pmatrix} \tag{4}$$

where $\mathbf{D}_i = \sum_{j<i} \mathbf{E}_{ji} + 4\mathbf{E}_{ii} + \sum_{j>i} \mathbf{E}_{ij}$. Since the $\mathbf{E}_{ij}$ are chosen uniformly at random, we see that the matrix $\mathbf{E}$ is just a uniformly random symmetric matrix in $\mathbb{F}_{q^m}^{k \times k}$, so the probability that $\mathbf{E}$ is singular is bounded by $\frac{1}{q^m - 1}$ (Lemma 10). Since the $\mathbf{v}_i$ are chosen uniformly at random in $\mathbb{F}_q^{n-o} \times \{0\}^o$, they are linearly dependent with probability bounded by $\frac{q^{k-(n-o)}}{q-1}$ (Lemma 10 again), and otherwise the $\mathbf{M}_i$ are independent and uniformly random matrices. Equation (4) shows that if the $\mathbf{v}_i$ are linearly independent and $\mathbf{E}$ is nonsingular, then the $\mathbf{M}'_i$ are also uniformly random. Therefore, by Lemma 10, $\mathcal{P}'^\star(\mathbf{v}, \cdot)$ has full rank except with probability bounded by

$$\frac{1}{q^m - 1} + \frac{q^{k-(n-o)}}{q-1} + \frac{q^{m-ko}}{q-1}. \qquad \square$$

B Proof of Lemma 8

Proof. The EUF-KOA adversary $\mathcal{B}$ works as follows. When $\mathcal{B}$ is given a public key $\mathcal{P}$, it starts simulating $\mathcal{A}$ on input $\mathcal{P}$. To simulate random oracle queries $\mathcal{B}$ maintains a list of queries L, that is initially empty. When $\mathcal{A}$ queries a random oracle at input m, $\mathcal{B}$ responds with $(\mathbf{E}, \mathbf{t})$ if there is an entry $(m, \mathbf{E}, \mathbf{t}) \in L$ and otherwise $\mathcal{B}$ samples $\mathbf{E} = \{\mathbf{E}_{ij}\}_{1 \le i \le j \le k} \in \mathbb{F}_{q^m}$ and $\mathbf{t} \in \mathbb{F}_q^m$ uniformly at random, adds $(m, \mathbf{E}, \mathbf{t})$ to L and responds with $(\mathbf{E}, \mathbf{t})$.

When $\mathcal{A}$ makes a query to sign a message M, $\mathcal{B}$ chooses a random salt and aborts if there is an entry $(m||\mathsf{salt}, \star, \star)$ in L. Otherwise, $\mathcal{B}$ samples $\mathbf{E} = \{\mathbf{E}_{ij}\}_{1 \le i \le j \le k} \in \mathbb{F}_{q^m}$ and $\mathbf{s}_1, \ldots, \mathbf{s}_k \in \mathbb{F}_q^n$, and sets $\mathbf{t} = \sum_{ij} \mathbf{E}_{ij} \mathcal{P}(\mathbf{s}_i + \mathbf{s}_j)$. Then $\mathcal{B}$ adds $(m||\mathsf{salt}, \mathbf{E}, \mathbf{t})$ to L and outputs the signature $(\mathsf{salt}, \mathbf{s}_1, \cdots, \mathbf{s}_k)$.

Finally, when $\mathcal{A}$ outputs a message-signature pair (m, σ), $\mathcal{B}$ just outputs the same pair.

It is clear that $\mathcal{B}$ runs in time $T + O((Q_h + Q_s + 1)\mathrm{poly}(n, m, k, q))$, so to finish the proof we need to show that $\mathcal{B}$ succeeds in the EUF-KOA game with a sufficiently large probability. We prove this with a sequence of games.

- Let Game_0 be $\mathcal{A}$'s EUF-CMA game against the MAYO signature scheme. By definition we have $\Pr[\mathsf{Game}_0() = 1] = Adv^{\mathsf{EUF-CMA}}_{n,m,o,k,q}(\mathcal{A})$.
- Let Game_1 be identical to Game_0, except that the game aborts and outputs 0 if to answer a signing query m, the challenger picks a salt, such that the random oracle was already queried at input $m||\mathsf{salt}$. Since there are in total $Q_h + Q_s$ queries to the random oracle, the probability of an abort is at most $(Q_s + Q_h)2^{-2\lambda}$ for each signing query, which makes for a total probability of an abort of $(Q_s + Q_h)Q_s 2^{-2\lambda}$. Therefore, we have $\Pr[\mathsf{Game}_1() = 1] \ge \Pr[\mathsf{Game}_0() = 1] - (Q_s + Q_h)Q_s 2^{-2\lambda}$.
- Let Game_2 be the same as Game_1 except that the game aborts and outputs 0 if during one of the calls to the signing oracle, the challenger has to restart the signing algorithm because he arrives at a linear system $\mathcal{P}^\star(\mathbf{v}_1 + \mathbf{o}_1, \ldots, \mathbf{v}_k + \mathbf{o}_k) = \mathbf{t}$ which does not have full rank. Note that the view of the adversary in Game_1 is independent of the number of signing attempts: if the signing algorithm encounters a system that does not have full rank, it just restarts from the beginning. Therefore, the output of the signing algorithm is independent of the number of signing attempts. It follows from Lemma 3 that

$$\begin{aligned}\Pr[\mathsf{Game}_2() = 1] &= \Pr[\mathsf{Game}_1() = 1 \wedge \mathsf{no\ restart}] = \Pr[\mathsf{Game}_1() = 1]\Pr[\mathsf{no\ restart}] \\ &\ge \Pr[\mathsf{Game}_1() = 1]\left(1 - Q_s\left(\frac{1}{q^m - 1} + \frac{q^{k-(n-o)}}{q-1} + \frac{q^{m-ko}}{q-1}\right)\right).\end{aligned}$$

- The final game Game_3 is just the EUF-KOA game played by $\mathcal{B}^{\mathcal{A}}$. If Game_2 does not abort, then the view of $\mathcal{A}$ is identical in Game_2 and Game_3, because if no salt is chosen more than once for the same message, then $\mathcal{B}$ simulates the random oracle perfectly. Moreover, since all of the linear systems have full rank, the signatures are computed as $\mathbf{s} = \mathbf{v} + \mathbf{o}$, where $\mathbf{v}$ is chosen uniformly at

random in $(\mathbb{F}_q^{n-o} \times \{0\}^o)^k$, and $\mathbf{o}$ is uniformly random in O^k. By construction we have $(\mathbb{F}_q^{n-o} \times \{0\}^o) + O = \mathbb{F}_q^n$, so the signatures in Game_2 are uniformly distributed, which means that $\mathcal{B}$ simulates the signing oracle perfectly by just choosing random $\mathbf{s} \in \mathbb{F}_q^{kn}$. Therefore, the probability that $\mathcal{A}$ outputs a forgery in Game_2 is at least as big as the probability that it outputs a forgery in Game_3 (it could be larger, since Game_3 aborts less often, but this is not important for our analysis), so we have $\Pr[\mathsf{Game}_3() = 1] > \Pr[\mathsf{Game}_2() = 1]$.

By combining the 3 inequalities we get that

$$\mathsf{Adv}^{\mathsf{EUF-CMA}}_{n,m,o,k,q}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{EUF\text{-}KOA}}_{n,m,o,q}(\mathcal{B}) \left(1 - Q_s \left(\frac{1}{q^m - 1} + \frac{q^{k-(n-o)}}{q-1} + \frac{q^{m-ko}}{q-1}\right)\right)^{-1} + (Q_h + Q_s) Q_s 2^{-2\lambda}. \qquad \square$$

C Proof of Lemma 9

Proof. We do the proof with a short sequence of games. The first game Game_0 is the EUF-KOA game played by $\mathcal{A}$. By definition we have $\Pr[\mathsf{Game}_0() = 1] = \mathsf{Adv}^{\mathsf{EUF\text{-}KOA}}_{n,m,o,k,q}(\mathcal{A})$.

The next game is the same as Game_0, except that during the key generation step the challenger chooses a uniformly random $\mathcal{P} \in \mathsf{MQ}_{n,m,q}$, instead of a $\mathcal{P}$ that vanishes on some oil space O. We construct the adversary $\mathcal{B}$ against the UOV assumption as follows. When $\mathcal{B}$ is given a multivariate quadratic map $\mathcal{P}$, it computes the matrix representation $\{\mathbf{P}_i^{(1)}, \mathbf{P}_i^{(2)}, \mathbf{P}_i^{(3)}\}_{i \in [m]}$ of $\mathcal{P}$. Then, $\mathcal{B}$ pick a random seed, and runs $\mathcal{A}$ on input $\mathsf{pk} = (\mathsf{seed}, \{\mathbf{P}_i^{(3)}\}_{i \in [m]})$, while faithfully simulating a random oracle, and an Expand oracle that outputs $\mathbf{P}_i^{(1)}$ on input $\mathsf{seed}||\mathsf{P1}||i$, that outputs $\mathbf{P}_i^{(2)}$ on input $\mathsf{seed}||\mathsf{P1}||i$, and that outputs random matrices of the appropriate shape otherwise. We designed $\mathcal{B}$ in such a way, that if $\mathcal{B}$ is given a $\mathcal{P}$ that is a (n, m, o, q) UOV map, then $\mathcal{B}$ is exactly Game_0, and if $\mathcal{B}$ is given a random map $\mathcal{P}$, then $\mathcal{B}$ is Game_1. Therefore we have

$$\mathsf{Adv}^{\mathsf{UOV}}_{n,m,o,q}(\mathcal{B}) = |\Pr[\mathsf{Game}_0() = 1] - \Pr[\mathsf{Game}_1() = 1]|\,.$$

For the next game we define the adversary $\mathcal{B}'$ against the whipped MQ problem. When $\mathcal{B}'$ is given a WMQ instance $\mathcal{P}, \{\mathbf{E}_{ij}\}_{ij}, \mathbf{t}$, it does the same thing as Game_1, except that instead of simulating a random oracle honestly, $\mathcal{B}'$ chooses an integer $I \in [Q_h]$ uniformly at random, and outputs $(\{\mathbf{E}_{ij}\}_{ij}, \mathbf{t})$ for the I-th distinct random oracle query (and all the subsequent queries for the same message). If $\mathcal{A}$ outputs a valid message-signature pair $(m, (\mathsf{salt}, \mathbf{s}))$, then the $\mathcal{B}'$ adversary checks if $m||\mathsf{salt}$ was the I-th random oracle query. If this is the case, then $\mathcal{B}'$ outputs $\mathbf{s}$, which is a correct solution to the WMQ instance, and otherwise $\mathcal{B}'$ aborts. The view of $\mathcal{A}$ in this game is the same as the view of a in Game_1, so $\mathcal{A}$ outputs a valid message-signature pair with probability $\Pr[\mathsf{Game}_1() = 1]$. The probability that $\mathcal{A}$ outputs a valid pair $(m, (\mathsf{salt}, \mathbf{s}))$ such that it has not queried the random oracle on input $m||\mathsf{salt}$ is at most q^{-m}. Note that the guess I is

information-theoretically hidden from $\mathcal{A}$, so if $\mathcal{A}$ outputs a valid forgery for the J-th random oracle query, then the probability that $I = J$ is $1/Q_h$. Therefore we have $\mathsf{Adv}^{\mathsf{WMQ}}_{n,m,k,q}(\mathcal{B}') \geq (\Pr[\mathsf{Game}_1() = 1] - q^{-m})/Q_h$.

We can now finish the proof by combining $\Pr[\mathsf{Game}_0() = 1] = \mathsf{Adv}^{\mathsf{EUF\text{-}KOA}}_{n,m,o,k,q}(\mathcal{A})$ with inequalities from the two game transitions to get

$$\mathsf{Adv}^{\mathsf{EUF\text{-}KOA}}_{n,m,o,k,q}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{UOV}}_{n,m,o,q}(\mathcal{B}) + Q_h \mathsf{Adv}^{\mathsf{WMQ}}_{n,m,k,q}(\mathcal{B}') + q^{-m}.$$

□

References

1. Beullens, W.: Improved cryptanalysis of UOV and rainbow. Cryptology ePrint Archive, Report 2020/1343 (2020). https://eprint.iacr.org/2020/1343
2. Beullens, W.: Sigma protocols for MQ, PKP and SIS, and fishy signature schemes. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020. LNCS, vol. 12107, pp. 183–211. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-45727-3_7
3. Chen, M.-S., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 135–165. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_5
4. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_27
5. 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., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 242–257. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_15
6. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, pp. 75–83 (2002)
7. 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). https://doi.org/10.1007/3-540-48910-X_15
8. 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). https://doi.org/10.1007/BFb0055733
9. Lyubashevsky, V., et al.: Crystals-Dilithium. Technical report, National Institute of Standards and Technology (2020). https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions
10. Petzoldt, A., Thomae, E., Bulygin, S., Wolf, C.: Small public keys and fast verification for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic public key systems. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 475–490. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23951-9_31
11. Prest, T., et al. FALCON. Technical report, National Institute of Standards and Technology (2020). https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

12. Samardjiska, S., Chen, M.-S., Hulsing, A., Rijneveld, J., Schwabe, P.: MQDSS. Technical report, National Institute of Standards and Technology (2019). https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions
13. 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). https://doi.org/10.1007/978-3-642-30057-8_10
14. Wagner, D.: A generalized birthday problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–304. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_19