MQ-Sign: A New Post-Quantum Signature Scheme based on Multivariate Quadratic Equations: Shorter and Faster *

Kyung-Ah Shim[1] and Hyeokdong Kwon[1]

National Institute for Mathematical Sciences
kashim,hyeokdong@nims.re.kr

Abstract. We present the digital signature scheme, MQ-Sign, based on UOV whose security relies on the hardness of solving large systems of multivariate quadratic equations. MQ-Sign supports two types of signature schemes, MQ-Sign-RR and MQ-Sign-LR, depending on the selection of Vinegar∗Vinegar quadratic terms. MQ-Sign-LR uses the linear combinations of v lines and Vinegar variables as Vinegar∗Vinegar quadratic terms which provides a compact representation as a circulant matrix-vector product. This structure reduces the secret key size significantly and improve the performance of key generation and signing. For faster signing performance, it also uses the block inversion method based on half-sized block matrices and Schur complement. Compared to MQ-Sign-RR, the secret key of MQ-Sign-LR is reduced by about 42% and performance of key generation and signing of MQ-Sign-LR is improved by 30% It is designed to enable dramatically faster online signing by precomputing most of heavy part for solving the linear system. Finally, MQ-Sign with precomputation is 4.6x to 6.3x faster than the scheme without precomputation at the three security levels.

Keywords: Block matrix inversion · Multivariate quadratic equation · UOV.

1 Introduction

Multivariate quadratic equations(MQ)-based signature schemes are mainly based on the hardness of solving large systems of multivariate quadratic equations, called MQ-problem. In MQ-schemes, a trapdoor is hidden in secret affine layers using the affine-substitute-affine (ASA) structure. Security of this ASA structure relies on the hardness of variants of Extended Isomorphism of Polynomials (EIP) problem [23]. Unbalanced Oil and Vinegar (UOV) signature scheme is one of the oldest and best studied cryptosystems. Rainbow, a variant of UOV, is based on the multiple-layered structure to reduce the key size and improve performance [15]. These MQ-signature schemes are very simple and fast, and has small signatures. Since they require simple operations such as matrix-vector products

* This work is submitted to 'Korean Post-Quantum Cryptography Competition' (www.kpqc.or.kr).

and solving linear systems over small finite fields, they can be efficiently implemented on resource-constrained devices [9, 11, 12]. Recent advanced attacks on Rainbow [38, 2, 3, 33, 6, 7] made UOV a better choice both in terms of security and efficiency. Although NIST recommended three algorithms, Dilithium, Falcon and SPHINCS+ as digital signature schemes in PQC Standardization. To diversify the signature portfolio, NIST received proposals for signature schemes with short signatures and fast verification that are not based on structured lattices. The MQ-signature scheme such as UOV is emerging as a strong candidate. In this document, we propose an efficient MQ-signature scheme, MQ-Sign, based on UOV with shorter secret key size and faster performance.

1.1 Design rationale and Advantages

Our scheme is designed with the following design rationale.

A New MQ-Signature Scheme based on UOV. MQ-Sign is based on UOV. UOV has withstood rigorous security analysis for a long time since its invention 1999. It is older, simpler, and has a strictly smaller attack surface in comparison to Rainbow. MQ-Sign maintains the structure of UOV and provides shorter secret key size and faster performance. It provides MQ-Sign-RR and MQ-Sign-LR as follows:

- MQ-Sign-RR uses random Vinegar∗Vinegar indexed quadratic terms and random Vinegar∗Oil quadratic terms as the central map $\mathcal{F}$ and the equivalent key of the form $\mathcal{T} = \begin{pmatrix} I & T \\ 0 & I \end{pmatrix}$ as the linear map.
- MQ-Sign-LR uses the linear combinations of v lines and Vinegar variables as Vinegar∗Vinegar quadratic terms in the central map $\mathcal{F}$. It provides a compact representation of Vinegar∗Vinegar quadratic terms as a circulant matrix-vector product. This structure reduces the secret key size significantly and improve the performance of key generation and signing.

Security Guarantee against Potential Attacks. In order to prevent potential attacks, we use a binding technique so that a signature is identified with a unique public key and message.

- For given two public keys $\mathcal{P}$ and $\mathcal{P}'$ such that $\mathcal{P}' = \mathcal{P} \circ T'$, if $\sigma = (\mathbf{z}, \mathbf{r})$ is a signature on a message M under the public key $\mathcal{P}$ then one who knows T' can generate a valid signature $\sigma' = (\mathbf{z}', \mathbf{r})$ on the same message M under the public key $\mathcal{P}'$ by computing $\mathbf{z}' = (\mathbf{T}')^{-\mathbf{1}}(\mathbf{z})$.
- To prevent this kind of attacks, one needs to bind a message being signed with the public key, i.e. $H(M||r||H(\mathcal{P}))$. So, we use $H(M||r||H(\mathcal{P}))$ in the signing and verification algorithms.

Small Signatures and Shorter Secret Keys. Like other MQ-schemes, the signatures of MQ-Sign are very small. More precisely, the signature sizes of MQ-Sign- require 150 bytes, 216 bytes and 276 bytes at security levels 1, 3, and

5, respectively. Compared to UOV, the public key size of MQ-Sign is slightly reduced. The compact representation of Vinegar∗ Vinegar quadratic terms in the secret key of MQ-Sign-LR reduces the secret key size by about 37% compared to MQ-Sign-RR.

Fast Performance and Easy to Implement Like many other MQ-signature schemes, MQ-Sign is very simple and is easy to implement. Simple operations such as matrix-vector products, solving linear systems for small finite fields are required, and fast signing and verification are possible. In Rainbow with two layers, the number of equations is divided into two which reduces the size of the matrix being inverted. Since MQ-Sign has a single layer, it requires relatively large size of the matrix in Gaussian elimination which makes signing inefficient.

- In order to resolve this inefficiency, we use the block inversion method proposed in [32] that exploits the inversions of half-sized matrices and Schur complement. So, MQ-Sign provides faster signing performance.
- Due to the simple generation of Vinegar∗ Vinegar quadratic terms based on lines and the computation of a circulant matrix-vector product for the Vinegar value substitution, the performance of key generation and signing of MQ-Sign-LR is improved by about 30% to 50% compared to MQ-Sign-RR.
- MQ-Sign is designed to enable dramatically faster online signing by precomputing most of heavy part for solving the linear system. Finally, MQ-Sign with precomputation is 4.6x to 6.3x faster than the scheme without precomputation at the three security levels.

Protection Side-Channel Attacks. All key dependent operations in our scheme are performed in a constant-time manner. MQ-Sign can prevent the side-channel attack in [22] due to its single layer structure.

1.2 Limitations

Large Key Sizes. Despite the short signature size and fast performance, the MQ-schemes suffer from relatively large public/secret key sizes. Even though the secret key size of MQ-Sign-LR is much smaller than that of UOV, it remains larger than the size of some other post-quantum signature schemes. However, due to increasing memory capabilities even of medium devices (e.g. smartphones), we do not think that this will be a major problem.

2 Our Signature Scheme: MQ-Sign

2.1 Basic Operations

Main Parameters.

- $\mathbb{F}_q$: a finite field of q elements
- m: the number of polynomials in the public key
- v: the number of Vinegar variables
- o: the number of Oil variables, $m = o$
- n: the number of variables in the public key, $n = o + v$.

Let $V = \{1, \cdots, v\}$ and $O = \{v+1, \cdots, v+o\}$ be sets of integers such that $|V| = v$, $|O| = o$, and $n = v + o$. We first describe the structure of UOV [20]. A central map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^o$ of UOV, $\mathcal{F} = (\mathcal{F}^{(1)}, \cdots, \mathcal{F}^{(o)})$, is o multivariate quadratic equations with n variables $x_1, \cdots, x_n$ defined by

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

Each polynomial $\mathcal{F}^{(k)}$ has no quadratic terms indexed by Oil∗Oil, i.e. the quadratic terms $x_i x_j$ for $i, j \in O$. This is called the missing Oil∗Oil structure that allows to invert the quadratic systems in signing. An invertible affine map $\mathcal{T} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is required to destroy the missing Oil∗Oil structure of $\mathcal{F}$. A public key is $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ that seems to be hardly distinguishable from a random quadratic system, thus be hard to invert, where $(\mathcal{F}, T)$ is a secret key.

Each central quadratic polynomial $\mathcal{F}^{(k)}$ is written as

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

where $\mathcal{F}_V^{(k)}$ and $\mathcal{F}_{OV}^{(k)}$ are the part of Vinegar×Vinegar quadratic terms and the part of Vinegar×Oil quadratic terms, respectively, and $\mathcal{F}_L^{(k)}$ and $\mathcal{F}_C^{(k)}$ are the part of linear terms and constant terms, respectively, for $k = 1, \cdots, o$. In UOV, the central polynomial (1) can be written by

$$\mathcal{F}^{(k)} = \mathcal{F}_{V,R}^{(k)} + \mathcal{F}_{OV,R}^{(k)} + \mathcal{F}_L^{(k)} + \mathcal{F}_C^{(k)}.$$

2.2 Central Maps and Linear Maps

MQ-Sign provides MQ-Sign-RR and MQ-Sign-LR depending on the selection of the Vinegar8Vinegar quadratic terms.

[Selection of $\mathcal{F}_V^{(k)}$.]

– **Random$\mathcal{F}_V^{(k)}$.** The the Vinegar×Vinegar quadratic part, $\mathcal{F}_V^{(k)}$, is chosen as

$$\mathcal{F}_V^{(k)} = \mathcal{F}_{V,R}^{(k)} = \sum_{i \in O, j \in V} \alpha_{ij}^{(k)} x_i x_j,$$

where $\alpha_{ij}^{(k)}$ is selected randomly from $\mathbb{F}_q$ so that the symmetric matrix of the quadratic part of $\mathcal{F}_V^{(k)}$ has full rank for $k = 1, \cdots, o$.

– **$\mathcal{F}_V^{(k)}$ using v Lines.** For the Vinegar×Vinegar quadratic part, $\mathcal{F}_V^{(k)}$, is chosen as

$$\mathcal{F}_V^{(1)} = \mathcal{F}_{V,LR}^{(1)} = x_1 \cdot L_1 + x_2 L_2 + \cdots + x_v L_v,$$
$$\mathcal{F}_V^{(2)} = \mathcal{F}_{V,LR}^{(2)} = x_v \cdot L_1 + x_1 L_2 + \cdots + x_{v-1} L_v,$$
$$\cdots,$$
$$\mathcal{F}_V^{(o)} = x_{v-o_1+2} \cdot L_1 + x_{v-o_1+3} L_2 + \cdots + x_{v-o_1+1} L_v,$$

where $L_i = \sum_{j=1}^{v} \delta_j x_j$ $(1 \le i \le v)$ is a line in variables $(x_1, \cdots, x_v)$, randomly chosen in $\mathbb{F}_q^*$ so that the symmetric matrix of the quadratic part of $\mathcal{F}_{V,LR}^{(k)}$ has full rank for $k = 1, \cdots, o$. Then, the system of o polynomials with v variables can be represented as

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_v \\ x_v & x_1 & \cdots & x_{v-1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{v-o_1+2} & x_{v-o_1+3} & \cdots & x_{v-o_1+1} \end{pmatrix} \cdot \begin{pmatrix} L_1 \\ L_2 \\ \cdots \\ L_v \end{pmatrix}.$$

The substitution of the random Vinegar values into the Vinegar×Vinegar quadratic terms can be computed as the following circulant matrix-vector product

$$\begin{pmatrix} \mathcal{F}_{V,LR}^{(1)}(s_V) \\ \mathcal{F}_{V,LR}^{(2)}(s_V) \\ \cdots \\ \mathcal{F}_{V,LR}^{(o)}(s_V) \end{pmatrix} = \begin{pmatrix} x_1 & x_2 & \cdots & x_v \\ x_v & x_1 & \cdots & x_{v-1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{v-o_1+2} & x_{v-o_1+3} & \cdots & x_{v-o_1+1} \end{pmatrix} \cdot \begin{pmatrix} L_1(s_V) \\ L_2(s_V) \\ \cdots \\ L_v(s_V) \end{pmatrix},$$

where $s_V = (s_0, \cdots, s_v) \in \mathbb{F}_q^v$ is a vector of random Vinegar values. This construction reduces the size of the quadratic part $\mathcal{F}_V^{(k)}$ from $(v \times v)/2 \cdot o$ field elements in random $\mathcal{F}_V^{(k)}$ to $v \cdot (v + o) = v \cdot n$ field elements.

[**Selection of $\mathcal{F}_{OV}^{(k)}$.**] For the Vinegar×Oil quadratic part, $\mathcal{F}_{OV}^{(k)}$, is chosen as

$$\mathcal{F}_{OV}^{(k)} = \mathcal{F}_{OV,R}^{(k)} = \sum_{i,j \in V, i \le j} \beta_{ij}^{(k)} x_i x_j,$$

where $\beta_{ij}^{(k)}$ is selected randomly from $\mathbb{F}_q$ so that the symmetric matrix of the quadratic part of $\mathcal{F}_{OV}^{(k)}$ has full rank for $k = 1, \cdots, o$.

[**Selection of Linear Maps.**] The affine map is chosen as a linear map of the form $\mathcal{T}_E = \begin{pmatrix} I & T \\ 0 & I \end{pmatrix}$.

[**MQ-Sign-RR and MQ-Sign-LR.**] According to the selections of $\mathcal{F}_V^{(k)}$ and $\mathcal{F}_{OV}^{(k)}$ $(k = 1, \cdots, o)$, MQ-Sign-RR and MQ-Sign-LR are determined by the following two combinations for central maps and linear maps:

- Random $\mathcal{F}_V^{(k)}$, random $\mathcal{F}_{OV}^{(k)}$, and the linear map $\mathcal{T}_E$ for MQ-Sign-RR:

$$\mathcal{F}_{RR}^{(k)} = \mathcal{F}_{V,R}^{(k)} + \mathcal{F}_{OV,R}^{(k)}.$$

- Line-based $\mathcal{F}_{V,LR}^{(k)}$, random $\mathcal{F}_{OV,R}^{(k)}$, and the linear map $\mathcal{T}_E$ for MQ-Sign-LR:

$$\mathcal{F}_{LR}^{(k)} = \mathcal{F}_{V,LR}^{(k)} + \mathcal{F}_{OV,R}^{(k)}.$$

MQ-Sign has neither linear terms and constant terms.

MQ-Sign uses the equivalent key of the form $\mathcal{T}_E$. The public key $\mathcal{P} = (\mathcal{P}^{(1)}, \cdots, \mathcal{P}^{(m)})$ is computed by evaluating $\mathcal{P}^{(k)} = \mathcal{T}_E^{\mathtt{T}} \cdot \mathcal{F}^{(k)} \cdot \mathcal{T}_E$ from $\mathcal{F} = (\mathcal{F}^{(1)}, \cdots, \mathcal{F}^{(m)})$ and bringing the resulting matrices to upper triangular form. Since the blocks $\mathcal{F}^{(k)}$ are already upper triangular matrices, this operation has no impact on them. From

$$\begin{pmatrix} I & 0 \\ T & I \end{pmatrix} \begin{pmatrix} \mathcal{F}_1^{(k)} & \mathcal{F}_2^{(k)} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & T \\ 0 & I \end{pmatrix}$$

$$= \begin{pmatrix} \mathcal{F}_1^{(k)} & \mathcal{F}_1^{(k)} T + \mathcal{F}_2^{(k)} \\ T^{\mathtt{T}} \mathcal{F}_1^{(k)} & T^{\mathtt{T}} \mathcal{F}_1^{(k)} T + T^{\mathtt{T}} \mathcal{F}_2^{(k)} \end{pmatrix}$$

we deduce

$$\mathcal{P}^{(k)} = \begin{pmatrix} \mathcal{P}_1^{(k)} & \mathcal{P}_2^{(k)} \\ 0 & \mathcal{P}_3^{(k)} \end{pmatrix} = \begin{pmatrix} \mathcal{F}_1^{(k)} & (\mathcal{F}_1^{(k)} + \mathcal{F}_1^{(k)})^{\mathtt{T}} T + \mathcal{F}_2^{(k)} \\ 0 & \mathtt{Upper}(T^{\mathtt{T}} \mathcal{F}_1^{(k)} T + T^{\mathtt{T}} \mathcal{F}_2^{(k)}) \end{pmatrix}.$$

2.3 Bind Signatures with the Public Key

In order to prevent potential attacks, we use a binding technique so that a signature is identified with a unique public key and a message. Assume that there are two public keys $\mathcal{P}$ and $\mathcal{P}'$ such that $\mathcal{P}' = \mathcal{P} \circ T'$, where

$$\mathcal{P} = \mathcal{F} \circ T, \quad \mathcal{P}' = (\mathcal{F} \circ T) \circ T'.$$

If $\sigma = (\mathbf{z}, r)$ is a signature on a message M under the public key $\mathcal{P}$ then one who knows T' can generate a valid signature $\sigma' = (\mathbf{z}', r)$ on the same message M under the public key $\mathcal{P}'$ by computing $\mathbf{z}' = (T')^{-1}(\mathbf{z})$. This is similar to rogue-key attacks on aggregate or multisignature schemes in the multiuser setting [8,

10]. For a given signature on a message M under $\mathcal{P}$, another signature can be produced on the same message under $\mathcal{P}'$ related to $\mathcal{P}$. It is different from malleable signature scheme: if, on input a message and a signature under a public key, it is possible to efficiently compute a signature on a related message under the same public key. To prevent this type of attacks, it needs to bind a message being signed with the hash value of the public key, i.e. $H(M||r||H(\mathcal{P}))$. So, we use $H(M||r||H(\mathcal{P}))$ in the signing and verification algorithms. Consequently, a given signature can be identified with a unique public key and a message.

2.4 Solving Linear Systems

A main idea to invert a system of quadratic equations in the MQ-schemes is to convert the quadratic system to a linear system by substituting random Vinegar values into the Vinegar variables of the central polynomials. There are two major computations in signing.

- **Substitution of Vinegar Values into the Central Map.** Calculations for substituting the random Vinegar values into the central map are required. Since there are a large number of quadratic terms with Vinegar×Vinegar indexes and Vinegar×Oil indexes, the computations are heavy.
- **Solving Linear System.** After the Vinegar value substitution, Gaussian elimination is used to find a solution of the linear system, whose complexity is $O(o^3)$ for the number of equations o.

These computations are main bottlenecks for signing cost. Unlike Rainbow with two layers, UOV with a single layer is required to find a solution of relatively large linear system: UOV requires the inversion of an $o \times o$ matrix, where o is up to twice as large as o_i $(i = 1, 2)$ in Rainbow, where o_1 and o_2 are the numbers of equations in the first and second layers of Rainbow, respectively. In order to resolve this inefficiency, we use the block inversion method that exploits the inversions of half-sized matrices [32].

Block Matrix Inversion Method. In signing, UOV and Rainbow use Gaussian elimination to solve the linear system. In Rainbow implementation [29], the signing algorithm computes R^{-1} by using Gaussian elimination, where R is the coefficient matrix of the linear system obtained from substituting the Vinegar values. We use a fast method, the block matrix inversion (BMI) method proposed in [32] that computes $R^{-1} \cdot \alpha$ directly, without finding R^{-1}, by using the inversions of half-sized matrices.

- **BMI Method.** A nonsingular square matrix R of 2×2 blocks is represented by the LDU decomposition of block matrices based on the Schur complement as

$$R = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & O \\ CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & O \\ 0 & D - CA^{-1}B \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix} = L \cdot D_{Sc} \cdot U,$$

where $A_{Sc} = [D - CA^{-1}B]$ is the Schur complement of A. Thus, $R^{-1} \cdot \xi$ can be expressed by A^{-1} and the inverse of the Schur complement of A, $[D - CA^{-1}B]^{-1}$, if they exist,

$$R^{-1} \cdot \begin{pmatrix} \xi_1 \\ \cdots \\ \xi_o \end{pmatrix} = \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix} \begin{pmatrix} A^{-1} & O \\ 0 & [D - CA^{-1}B]^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -CA^{-1} & I \end{pmatrix} \begin{pmatrix} \xi_1 \\ \cdots \\ \xi_o \end{pmatrix}.$$

After computing A^{-1}, CA^{-1}, $C(A^{-1}B)$, $A_{Sc}^{-1} = [D - CA^{-1}B]^{-1}$ and $A^{-1}B$ via two inversions and four matrix multiplications of $o/2 \times o/2$ block matrices, all remaining computations are made by four block matrix-vector products as:

$$CA^{-1} \cdot (\xi_1, \cdots, \xi_{o/2})^{\mathtt{T}} + (\xi_{o/2+1}, \cdots, \xi_o)^{\mathtt{T}} = (\alpha_{o/2+1}, \cdots, \alpha_o)^{\mathtt{T}},$$

$$A^{-1} \cdot (\xi_1, \cdots, \xi_{o/2})^{\mathtt{T}} = (\beta_1, \cdots, \beta_{o/2})^{\mathtt{T}},$$

$$A_{Sc}^{-1} \cdot (\alpha_{o/2+1}, \cdots, \alpha_o)^{\mathtt{T}} = (\beta_{o/2+1}, \cdots, \beta_o)^{\mathtt{T}},$$

$$(\beta_1, \cdots, \beta_{o/2})^{\mathtt{T}} + (-A^{-1}B) \cdot (\beta_{o/2+1}, \cdots, \beta_o)^{\mathtt{T}} = (\gamma_1, \cdots, \gamma_{o/2})^{\mathtt{T}}.$$

Finally, $s_O = (\gamma_1, \cdots, \gamma_{o/2}, \beta_{o/2+1}, \cdots, \beta_o)$ is the solution of $R \cdot \mathbf{x} = \xi$, i.e. $R^{-1} \cdot \xi = s_O^{\mathtt{T}}$.

– **Repeated BMI.** The BMI method can be applied to these two half-sized matrices which results in four inversions of $o/4 \times o/4$ matrices and extra operations. Like this, for $o = 2^l \cdot o'$, it can be applied l times, where the number of these iterations of the BMI is defined as a depth. However, l iterations will not always be effective, because 2^l inversions of $o/2^l \times o/2^l$ matrices are required.

According to the results using the BMI method in [32], the larger the size of a matrix being inverted, the greater the performance improvement and the higher the security level, the greater the effect of the optimizations. We use the BMI method with depth 1 to solve the linear system in signing.

2.5 Precomputation

MQ-Sign is designed to allow most heavy part for solving the linear system in signing to be precomputed in offline.

Precomputation. Signing can be divided into two parts: one is independent of messages being signed, the other depends on the messages. MQ-Sign has significantly large message independent operations in signing. Thus, the offline precomputation can dramatically improve signing.

[Offline Signing]

– Choose random Vinegar values $s_V = (s_1, \cdots, s_v) \in \mathbb{F}_q^v$.
– Substitute s_V into o the secret polynomials $\mathcal{F}^{(k)}$ $(1 \leq k \leq o)$, and get a $o \times o$ coefficient matrix $R = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ and a constant vector $\mathbf{c_V} = (c_1, \cdots, c_o) = (\mathcal{F}_V^{(1)}(s_V), \cdots, \mathcal{F}_V^{(o)}(s_V))$.
– Compute A^{-1}, CA^{-1}, $CA^{-1}B$, A_{Sc}^{-1} and $A^{-1}B$. If A or A_{Sc} is not invertible then go back to the first step.
– Choose a random salt r.
– Store the precomputed values $< s_V, \mathbf{c_V}, A^{-1}, CA^{-1}, A_{sc}^{-1}, A^{-1}B, r >$.

[Online Signing]

– Compute $\mathbf{h} = H(M||r||ph)$ ($H(M||r)$ in LR) for a message M.
– Compute $R^{-1} \cdot \xi = s_O$ by computing four block matrix-vector products from the precomputed values in the BMI method, where $\xi = \mathbf{h} - c_V = (h_1 - c_1, \cdots, h_o - c_o)$ and $\mathbf{h} = (h_1, \cdots, h_o)$.
– Compute $\mathbf{z} = \begin{pmatrix} s_V^{\mathsf{T}} + T \cdot s_O^{\mathsf{T}} \\ s_O \end{pmatrix}$.
– Output $\sigma = (\mathbf{z}, \mathbf{r})$ as a signature on M.

MQ-Sign with precomputation is at most 6x faster than the original version without precomputation at the three security levels. According to the security analysis in [32], if some precomputed values together with signatures generated by them are exposed or reused then the secret key of the scheme is completely recovered. Thus, the precomputed values (actually, s_V) should be stored securely and should not be reused in signing.

2.6 Specifications of MQ-Sign

■ MQ-Sign

– **KeyGen**(1^λ)**.** For a security parameter λ, do the followings:
  • Choose $\mathcal{F}^{(k)}$ $(k = 1, \cdots, o)$ and $\mathcal{T}$ as follows:
    ∗ MQ-Sign-RR: $\mathcal{F}_{RR}^{(k)} = \mathcal{F}_{V,R}^{(k)} + \mathcal{F}_{OV,R}^{(k)}$ and $\mathcal{T}_E$.
    ∗ MQ-Sign-LR: $\mathcal{F}_{LR}^{(k)} = \mathcal{F}_{V,LR}^{(k)} + \mathcal{F}_{OV,R}^{(k)}$ and $\mathcal{T}_E$.
  • Output a public key as PK and a secret key as SK.
    ∗ MQ-Sign-RR: $PK =< \mathcal{P}, ph >$ and $SK =< \mathcal{F}, T, ph >$, where $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ and $ph = H(\mathcal{P})$.
    ∗ MQ-Sign-LR: $PK =< \mathcal{P} >$ and $SK =< \mathcal{F}, T >$.

– **Sign**(SK, λ, M)**.** Given a message M and a collision-resistant hash function $H : \{0,1\}^* \to \mathbb{F}_q^o$, compute $\mathbf{a} = \mathcal{F}^{-1}(\xi)$, i.e. $\mathcal{F}(\mathbf{a}) = \xi$ as follow:

  • **[Vinegar Value Substitution.]** Select Vinegar values $s_V = (s_1, \cdots, s_v) \in \mathbb{F}_q^v$ at random and do the followings:

  - ∗ Substitute s_V into o central polynomials $\mathcal{F}^{(k)}$ $(1 \leq k \leq o)$ and get o polynomials of unknowns $x_{v+1}, \cdots, x_{v+o}$ with an $o \times o$ coefficient matrix $R = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$.
  - ∗ Compute A^{-1}, CA^{-1}, $CA^{-1}B$, A_{sc}^{-1}, $A^{-1}B$.
  - ∗ If A or A_{sc} is not invertible, then choose another vector of Vinegar values s'_V and try again.

- **[Solving Linear System.]** Choose a l-bit random salt r,
  - ∗ MQ-Sign-RR: compute $\mathbf{h} = H(M||r||ph) \in \mathbb{F}_q^o$.
  - ∗ MQ-Sign-LR: compute $\mathbf{h} = H(M||r) \in \mathbb{F}_q^o$.

  Find a solution $s_O = (s_{v+1}, \cdots, s_{v+o})$ of the linear system $R \cdot \mathbf{x} = \xi$ by using the block matrices of the BMI method, where $\xi = \mathbf{h} - c_V$ and $c_V = (\mathcal{F}_V^{(1)}(s_V), \cdots, \mathcal{F}_V^{(o)}(s_V))$.

- **[Output.]** Compute $\mathbf{z} = \begin{pmatrix} s_V^{\mathsf{T}} + T \cdot s_O^{\mathsf{T}} \\ s_O \end{pmatrix}$. Output $\sigma = (\mathbf{z}, r)$ as a signature on M.

– **Verify**(PK, M, σ)**.** Given a signature $\sigma = (\mathbf{z}, r)$ on a message M and the public key $\mathcal{P}$,

- MQ-Sign-RR: check the equality $\mathcal{P}(\mathbf{z}) = H(M||r||ph)$.
- MQ-Sign-LR: check the equality $\mathcal{P}(\mathbf{z}) = H(M||r)$.

If the equality holds, output *valid.*

The KeyGen, Sign, and Verify algorithms of MQ-Sign are presented in Algorithm 1, 2, and 3, respectively.

Algorithm 1 KeyGen(λ)

Require: parameters (q, v, o), length of salt l.
Ensure: key pair (sk, pk).
1: $M_T \leftarrow \texttt{Matrix}(q, v \times o)$
2: $T \leftarrow M_T$
3: $\mathcal{T} = \begin{pmatrix} I & T \\ 0 & I \end{pmatrix} \leftarrow \texttt{Equi}(T)$
4: $\mathcal{F} \leftarrow \texttt{MQmap}(q, v, o)$
5: $\mathcal{P} \leftarrow \mathcal{F} \circ \mathcal{T}$
6: $ph \leftarrow H(\mathcal{P})$
7: $sk \leftarrow (\mathcal{F},\ T,\ ph)$ $(sk \leftarrow (\mathcal{F},\ T)$ in LR)
8: $pk \leftarrow (\mathcal{P},\ ph)$ $(pk \leftarrow \mathcal{P}$ in LR)
9: **Return** $(sk,\ pk)$

Algorithm 2 Sign(sk, M)

Require: message M, secret key $(\mathcal{F}, T)$, length of the salt l.
Ensure: signature $\sigma = (\mathbf{z}, r) \in \mathbb{F}_q^n \times \{0,1\}^l$.

repeat
  $s_V = (s_1, ..., s_v) \leftarrow_R \mathbb{F}_q$
  $R = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \leftarrow \texttt{Coeffi}(\mathcal{F}^{(1)}(s_V), \cdots, \mathcal{F}^{(o)}(s_V))$
  $c_V \leftarrow (\mathcal{F}_V^{(1)}(s_V), \cdots, \mathcal{F}_V^{(o)}(s_V))$
  $E \leftarrow A^{-1}$
until $\exists\ A^{-1}$
repeat
  $(G, I, J) \leftarrow (CA^{-1},\ CA^{-1}B,\ [D - CA^{-1}B]^{-1})$
until $\exists\ [D - CA^{-1}B]^{-1}$
$K \leftarrow A^{-1}B$
$r \leftarrow \{0,1\}^l$
$\mathbf{h} \leftarrow H(M||r||ph)$ ($\mathbf{h} \leftarrow H(M||r)$ in LR)
$\xi \leftarrow \mathbf{h} - c_V$
$s_O = (s_{v+1}, ..., s_n) \leftarrow \texttt{BMI}(R^{-1} \cdot \xi,\ G,\ J,\ K)$
$\mathbf{z} \leftarrow \begin{pmatrix} s_V^{\mathsf{T}} + T \cdot S_O^{\mathsf{T}} \\ S_O^{\mathsf{T}} \end{pmatrix}$
$\sigma \leftarrow (\mathbf{z}, r)$
Return σ

Algorithm 3 Verify(pk, M, σ)

Require: message M, signature $\sigma = (\mathbf{z}, r) \in \mathbb{F}_q^n \times \{0,1\}^l$.
Ensure: boolean value **TRUE** or **FALSE**.

$\mathbf{h} \leftarrow H(M||r||ph)$ ($\mathbf{h} \leftarrow H(M||r)$ in LR)
h' $\leftarrow \mathcal{P}(\mathbf{z})$
if h' == h then
  return TRUE
else
  return FALSE
end if

3 Security Analysis

3.1 Existential Unforgeability

We first describe a computational hard problem and its hardness assumption.

Definition 1 (MQ-Problem). *Given a system* $\mathcal{P} = (\mathcal{P}^{(1)}, \cdots, \mathcal{P}^{(m)})$ *of* m *quadratic equations defined over* $\mathbb{F}_q$ *in variables* $x_1, \cdots, x_n$ *and* $\mathbf{y} = (y_1, \cdots, y_m) \in \mathbb{F}_q^m$*, find values* $\mathbf{x}' = (x_1', \cdots, x_n') \in \mathbb{F}_q^n$ *such that* $\mathcal{P}(\mathbf{x}') = \mathbf{y}$*:* $\mathcal{P}^{(1)}(\mathbf{x}') = y_1, \cdots, \mathcal{P}^{(m)}(\mathbf{x}') = y_m$*.*

Definition 2 (Extended Isomorphism of Polynomials (EIP) Problem). *Given a nonlinear multivariate system* $\mathcal{P}$ *such that* $\mathcal{P} = S \circ \mathcal{F} \circ T$ *for linear or affine maps* S *and* T*, and* $\mathcal{F}$ *belonging to a special class of nonlinear polynomial system* $\mathcal{C}$*, find a decomposition of* $\mathcal{P}$ *such that* $\mathcal{P} = S' \circ \mathcal{F}' \circ T'$ *for linear or affine maps* S' *and* T'*, and* $\mathcal{F}' \in \mathcal{C}$*.*

Definition 3 (UOV Problem). *Given* $(\mathcal{P}, y)$*, find a preimage* $\boldsymbol{z} \in \mathbb{F}_q^n$ *such that* $\mathcal{P}(\boldsymbol{z}) = y$*, where* $\mathcal{P}$ *is derived from* $(\mathcal{P}, \mathcal{F}, \mathcal{T}) \leftarrow \mathsf{GenUOVfunc}(1^\lambda)$ *and a challenge* $y \in \mathbb{F}_q^m$*.*

Definition 4. *An algorithm* $\mathcal{A}$ *has advantage* ϵ *to solve the UOV problem if*

$$Adv_{\mathcal{A}}(t) = Pr\left[\mathcal{A}(\mathcal{P}, y) = \boldsymbol{z} \mid \mathcal{P} \leftarrow \mathsf{GenUOVfunc},\ y \leftarrow \mathbb{F}_q^m\right] \geq \varepsilon.$$

If there is no algorithm $\mathcal{A}(t, \varepsilon)$ *that solves the UOV problem then we define that the UOV problem is* (t, ε)*-hard.*

Definition 5 (UOV Assumption). *The UOV function generator* GenUOVfunc *is* $(t(\lambda), \epsilon(\lambda))$*-secure if there is no inverting algorithm that takes as input* $\mathcal{P}$ *generated* $\mathcal{P} \leftarrow$ GenUOVfunc *and a challenge* $y \in \mathbb{F}_q^m$ *to find a preimage* $\boldsymbol{z} \in \mathbb{F}_q^n$ *such that* $\mathcal{P}(\boldsymbol{z}) = y$ *at* $t(\lambda)$ *time with probability at least* $\epsilon(\lambda)$*.*

In [30], in order to achieve existential unforgeability against adaptive chosen-message attacks (EUF-CMA) of UOV, the authors used a usual security proof for the Full-Domain-Hash scheme by modifying the signing algorithm to provide uniform distribution of the signatures. Their slightly modified UOV scheme is to use a random salt r as $H(M||r)$ instead of $H(M)$. Then the modified signature has the form $\sigma = (\mathbf{z}, r)$, where $\mathbf{z}$ is an original UOV signature. The existential unforgeability of MQ-Sign follows the security proof of the modified UOV in [30].

Theorem 1. *If the UOV problem is* (ε', t')*-hard then the modified UOV is* $(\varepsilon, t, q_H, q_s)$*-secure in the* EUF-CMA *game, where*

$$\varepsilon' \geq \varepsilon \cdot \frac{1 - (q_H + q_s)q_s 2^{-l}}{q_H + q_s + 1}, \quad t' \geq t + (q_H + q_s + 1)(t_{\mathcal{P}} + \mathcal{O}),$$

$t_{\mathcal{P}}$ *is running time to evaluate* $\mathcal{P}$ *and* l *is the length of a salt.*

3.2 Security Analysis and Cost Analysis against Known Attacks

Our scheme based on the missing Oil∗Oil structure for inverting the quadratic map uses the sparse polynomials for improving signing performance and reducing the secret key size. Our scheme is considered as special cases of UOV central map preserving full rank of the corresponding symmetric matrices. Security analysis of our scheme against known algebraic attacks is similar to those of UOV. We provide complexity estimates of our scheme against major algebraic attacks: direct attacks, Kipnis-Shamir attacks, key recovery attacks using good keys and intersection attacks. Throughout this document, we denote by the term 'complexity' the number of field multiplications an algorithm performs before outputting a solution. Our complexity estimates are expressed as the base 2 logarithm of this number.

[**Direct Attacks.**] The most straightforward way to cryptanalyze the MQ-signature schemes is to solve the public system $\mathcal{P}(x) = H(M||r||ph)$. The public keys behave like random systems and the degree of regularity of the system derived from the public key is the same as that of random systems of the same size. In order to solve the resulting quadratic system, the attacker can use an arbitrary method such as XL, Polynomial XL, Gröbner Basis algorithms and hybrid algorithms [4, 17]. The selection of o for our scheme depends on their security against the direct attacks. We summarize complexity of our scheme against the direct attacks at the three security levels using the known algorithms for solving the MQ-problem in Table 1, Table 2, and Table 3. According to this analysis, we choose $o \geq 46, 72, 96$ at the security levels 1, 3, and 5, respectively.

Table 1. Complexity estimates against the direct attacks at the security level 1.

Algorithms	44	46	48	50	52
Hybrid F5	131.86	137.43	142.99	148.56	154.12
Wiedemann XL	133.40	138.98	144.55	150.13	155.70
Polynomial XL	125.50	131.25	138.19	142.66	146.99

Table 2. Complexity estimates against the direct attacks at the security level 3.

Algorithms	68	70	72	74	76
Hybrid F5	195.37	200.92	203.58	209.03	214.59
Wiedemann XL	196.93	202.51	204.97	210.41	216.01
Polynomial XL	189.41	194.50	199.39	203.04	209.49

Table 3. Complexity estimates against the direct attacks at the security level 5.

Algorithms	94	96	98	100	102
Hybrid F5	261.18	266.50	272.10	277.32	279.90
Wiedemann XL	262.50	267.76	273.38	278.54	281.23
Polynomial XL	253.98	260.24	267.35	271.57	275.31

[Key Recovery Attacks using Good keys (UOV-Reconciliation).] The key recovery attacks using equivalent keys and good keys exploit the special structure of the central map, i.e. zero entries at certain known places to get equations with variables in $\mathcal{T}$. It is known that there exist a large number of different secret keys (called equivalent keys) for a given public key of the MQ-schemes [39, 37]. Wolf and Preneel [39] introduced the notion of equivalent keys as a fundamental tool to analyze the security of the MQ-schemes. Later, Thomae [37] generalized the notion of equivalent keys to good keys. If an adversary finds any of equivalent keys then the adversary can forge any signatures on any messages although it is not the same as the original secret key. For a private key $(\mathcal{F}, \mathcal{T})$, $(\mathcal{F}', \mathcal{T}')$ is an equivalent key of $(\mathcal{F}, \mathcal{T})$ if $\mathcal{P} = \mathcal{F} \circ \mathcal{T} = \mathcal{F}' \circ \mathcal{T}'$ and $\mathcal{F}'$ preserves all systematic zero coefficients of $\mathcal{F}$. Then, there is an equivalent key $(\mathcal{F}', \mathcal{T}')$ such that $\mathcal{P} = (\mathcal{F} \circ \Omega) \circ (\Omega^{-1} \circ \mathcal{T})$ of the secret key $(\mathcal{F}, \mathcal{T})$ with high probability such that

$$\mathcal{T}'^{-1} = \mathcal{T}^{-1} \cdot \Omega = \begin{pmatrix} I_{v\times v} & \widetilde{T'}_{v\times o} \\ 0_{o\times v} & I_{o\times o} \end{pmatrix}, \quad \Omega = \begin{pmatrix} \Omega^{(1)}_{v\times v} & 0_{v\times o} \\ \Omega^{(3)}_{o\times v} & \Omega^{(4)}_{o\times o} \end{pmatrix}. \tag{2}$$

To further decrease this complexity, the good keys are used, where the good keys do not preserve all the zero coefficients of $\mathcal{F}$, but just some of them. Thus, we can choose $\mathcal{F}$ and Ω more widely and further reduce the number of variables. The complexity of our scheme against the key recovery attacks using good keys is determined by solving a system of o quadratic equations with v variables:

$$Conplexity_{KRA}(q, o, v) = C_{MQ}(q, o, v),$$

where $C_{MQ}(q, o, v)$ denotes the complexity of solving a random system of o equations in v variables defined on $\mathbb{F}_q$ by using the algorithms for solving the MQ-problem.

[Kipnis-Shamir Attacks (UOV Attacks)]. The Kipnis-Shamir attacks were originally used to break the balanced Oil and Vinegar signature scheme ($v = o$) [20]. The attacks can be generalized to the unbalanced case ($v > o$). In the attacks, to find an equivalent key, we look for the space $\mathcal{T}^{-1}(\mathcal{O})$, where $\mathcal{O}$ is the Oil subspace of $\mathbb{F}_q^n$. Note that we get $P^{(i)} = T^{\mathsf{T}} \cdot F^{(i)} \cdot T$, where $F^{(i)}$ and $P^{(i)}$ are the symmetric matrices of the quadratic parts of $\mathcal{F}^{(i)}$ and $\mathcal{P}^{(i)}$, respectively, for $i = 1, \cdots, o$. Then the probability that the matrix $W_1^{-1} \cdot W_2$, where W_1 (invertible) and W_2 are random linear combinations of the matrices $P^{(i)}$ $(i = 1, \cdots, o)$, has a nontrivial invariant subspace (which is also a subspace of $\mathcal{T}^{-1}(\mathcal{O})$) is q^{v-o-1}. By computing the minimal invariant subspaces of $W_1^{-1} \cdot W_2$ and finding subspaces $\mathcal{T}^{-1}$ among them, the attack can recover the equivalent key. The complexity of the whole attack process is estimated by

$$Conplexity_{KS}(q, o, v) = q^{v-o-1} \cdot o^4.$$

[Intersection Attacks.] The intersection attack [6], an improved version of the Kipnis-Shamir attack, is considered as the most powerful attack among the

known attacks. Its complexity is

$$Conplexity_{Inter}(q, o, v) = C_{MQ}(q, ok(k+1)/2 - k(k-1), vk - o(k-1)),$$

where $k < v/(v-o)$. According to the complexity analysis of our schemes against the intersection attacks, we choose v such that $v > 1.5 \cdot o$. After determining the number of polynomials, o, we have to decide the number of Vinegar variables, v, depending on the Kipnis-Shamir attack, the key recovery attacks using good keys and the intersection attacks. Since the complexity of our schemes against the intersection attacks is lower than that of the reconciliation attacks, v can be determined by the intersection attacks.

[Replacement Attack.] One can mount some attacks by replacing the linear equation L_i with a new variable y_i, where $v > o$. Then the Vinegar×Vinegar quadratic parts of the secret key, $\mathcal{F}_V = (\mathcal{F}_V^{(1}, \cdots, \mathcal{F}_V^{(o)})$, are written in variables, $y_1, \cdots, y_v$, as

$$\mathcal{F}_V^{(1)} = \overline{L_1} \cdot y_1 + \overline{L_2} y_2 + \cdots + \overline{L_v} y_v,$$

$$\mathcal{F}_V^{(2)} = \overline{L_v} \cdot y_1 + \overline{L_1} y_2 + \cdots + \overline{L_{v-1}} y_v,$$

$$\cdots,$$

$$\mathcal{F}_V^{(o)} = \overline{L_{v-o+2}} \cdot y_1 + \overline{L_{v-o+3}} y_2 + \cdots + \overline{L_{v-o+1}} y_v,$$

where $\overline{L_i}$ is a line expressed by new variables, $y_1, \cdots, y_v$. This replacement preserves full rank of the symmetric matrix of the quadratic part of $\mathcal{F}_V^{(k)}$ for $k = 1, \cdots, o$.

Finally, we summarize complexities of our scheme against the attacks in Table 4, where $C_{MQ}(q, m, n)$ denotes the complexity of solving a random system of m equations in n variables defined on $\mathbb{F}_q$ by using the algorithms for solving the MQ-problem.

Table 4. Complexities of MQ-Sign(q, o, v) against the algebraic attacks.

Attack	Complexity
Direct Attack	$C_{MQ}(q,\ o,\ n)$
UOV-Reconciliation Attack	$C_{MQ}(q,\ o,\ v)$
Kipnis-Shamir Attack	$q^{v-o-1} \cdot o^4$
Intersection Attack	$C_{MQ}(1,\ ok(k+1)/2 - k(k-1),\ vk - o(k-1))$

[Implementation Attacks.] All key dependent operations in our scheme are performed in a constant-time manner to protect the timing attacks. MQ-Sign can prevent the side-channel attack in [22] due to the single layer structure. For secure implementations, the Vinegar values required in signing must not be revealed or reused [31, 1] .

3.3 Parameter Selection

Now, we suggest secure parameters at the three security levels in Table 5. Since the most powerful attacks all the attacks are the direct attack and the intersection attack, we give complexity estimates for the two major attacks in Table 5.

Table 5. Parameters and complexities of MQ-Sign(q, o, v) against the major attacks.

Security Level	1	3	5
(q, o, v)	$(\mathbb{F}_{2^8}, 46, 72)$	$(\mathbb{F}_{2^8}, 72, 112)$	$(\mathbb{F}_{2^8}, 96, 148)$
Direct(HF5)	135.5	202.4	262.3
Intersection attack	171.883	242.9	304.5

4 Computational Efficiency

4.1 Reference and AVX2 Optimized Implementations

We measure the performance of each algorithm on an Intel Xeon(R) Gold 6234 processor at the clock frequency of 3.3GHz. Reference and AVX2-optimized implementation results of MQ-Sign-RR and MQ-Sign-LR are given in Table 6. The results presented in Table 6 include the numbers of CPU cycles required by the key generation, signing and verification. Each result of signing and verification (resp. key generation) is a median of 10,000 (resp. 1,000) measurements. The source code was developed on Ubuntu 22.04 LTS, compiled using gcc 11.4.0, and optimization level -O3 applied. Hyperthreading and Turbo Boost are switched off. Signature and key sizes of MQ-Sign are given in Table 7.

Table 6. Performance of MQ-Sign.

Scheme	Security Level	1	3	5
Performance (Reference Code, median cycles)				
MQ-Sign-RR	KeyGen	122,046,651	438,023,770	994,466,810
	Sign	861,724	1,752,258	3,053,560
	Verify	755,522	1,339,244	2,218,340
MQ-Sign-LR	KeyGen	89,401,545	312,936,150	701,261,588
	Sign	451,262	1,004,830	2,026,304
	Verify	774,652	1,414,666	2,202,376
Performance (AVX2-optimized, median cycles)				
MQ-Sign-RR	KeyGen	9,454,708	40,250,626	102,775,550
	Sign	90,480	268,866	524,030
	Verify	50,460	185,086	363,611
MQ-Sign-LR	KeyGen	5,451,597	25,605,484	67,485,424
	Sign	65,300	168,684	360,636
	Verify	51,744	191,986	381,019

Table 7. Key/Signaute sizes of MQ-Sign.

Scheme	Security Level	1	3	5
MQ-Sign-RR	Public Key	328,505	1,238,825	2,893,025
	Secret Key	276,649	1,044,385	2,436,769
	Signature	150	216	276
MQ-Sign-LR	Public Key	328,441	1,238,761	2,892,961
	Secret Key	160,881	601,249	1,400,113
	Signature	150	216	276

MQ-Sign-RR and MQ-Sign-LR have the following differences: i) they have different secret keys, ii) MQ-Sign-RR includes the hash value of the public key ($ph = H(\mathcal{P})$) in the public/secret key and uses $H(M||r||ph)$ in signing, and iii) MQ-Sign-LR does not use ph at all. Consequently, they have different performance of key generation and signing, but their verification performance has little difference.They have different public/secret key sizes, but their signature

sizes remain unchanged. Compared to MQ-Sign-RR and UOV, the secret key of MQ-Sign-LR is reduced by about 42%. The performance of key generation and signing of MQ-Sign-LR is improved by 34% to 42% and 27% to 37% compared to MQ-Sign-RR. Despite fast performance of signing and verification, the key generation is slow. We believe our AVX2-optimized implementation of the key generation still has room for improvement.

4.2 Hashing of Public Key

For stronger security, the hash value of the public key is included in the public/secrt key of MQ-Sign-RR, which increases the public/secret key size slightly and makes key generation inefficient.

Actually, the computation of $H(\mathcal{P})$ is very heavy since the size of the public key is large. To mitigate this inefficiency, we can use the part of the public key as input to the hash function, i.e. $H(\mathcal{P}_2||\mathcal{P}_3)$, where $\mathcal{P}_2 = \{\mathcal{P}_2^{(k)}\}_{k=1}^{o}$ and $\mathcal{P}_3 = \{\mathcal{P}_3^{(k)}\}_{k=1}^{o}$. It is because that, in the rogue-key attack described in §2.4, $\mathcal{P}_1^{(k)} = \mathcal{P}_1'^{(k)}$ $(k = 1, \cdots, o)$ for the public keys $\mathcal{P}$ and $\mathcal{P}'$ when the linear maps $\mathcal{T}$ and $\mathcal{T}'$ are used as the form of equivalent keys. Actually, $\mathcal{P}$ and $\mathcal{P}'$ have the same central map $\mathcal{F}$, different linear maps $\mathcal{T}$ and $\mathcal{T} \circ \mathcal{T}'$, respectively, and $\mathcal{P}_1^{(k)}$ does not depend on the linear maps. Therefore, the attack can be prevented by removing $\mathcal{P}_1$ from the public key, where $\mathcal{P}_1 = \{\mathcal{P}_1^{(k)}\}_{k=1}^{o}$. In Table 8, MQ-Sign-RR(s) and MQ-Sign-RR(m) mean strong security and medium security against the potential attacks by including $H(\mathcal{P})$ and $H(\mathcal{P}_2||\mathcal{P}_3)$ in the public/secret key, respectively. MQ-Sign-RR(o) means not using the hash value of the public key.

Table 8. Performance of key generations for MQ-Sign.

Security Level	1	3	5
MQ-Sign-RR(s)	9,454,708	40,250,626	102,775,550
MQ-Sign-RR(m)	8,291,246	36,168,836	91,291,653
MQ-Sign-RR(o)	6,633,743	30,024,722	77,684,841

4.3 Precomputation

MQ-Sign is designed to allow most of the signature generation to be computed offline. Precomputation of most of heavy part for solving the linear system leads to dramatically faster online signing. This precomputation requires additional memory to store the precomputation values. The offline precompuations of MQ-Sign-RR and MQ-Sign-LR are different due to the use of different secret keys, but their on-line signing is the same. The performance of MQ-Sign with precomputation is given in Table 9. Finally, MQ-Sign with precomputation is 4.6x, 5.2x, and 6.3x faster than the scheme without precomputation at the three security levels, respectively.

Table 9. Performance of MQ-Sign with precomputation.

Scheme	Security Level	1	3	5
MQ-Sign-RR (MQ-Sign-LR)	Sign	19,532	49,256	84,465
	Memory	2,266	5,400	9,492

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, P. Briaud, M. Bros, P. Gaborit, V. Neiger, O. Ruatta, and J-P. Tillich, An algebraic attack on rank metric code-based cryptosystems, EUROCRYPT 2020, Part III, LNCS 12107, pp. 64–93, 2020.
3. M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. A. Perlner, D. Smith-Tone, J-P. Tillich, and J. A. Verbel, Improvements of algebraic attacks for solving the rank decoding and MinRank problems, ASIACRYPT 2020, Part I, LNCS 12491, pp. 507–536, 2020.
4. L. Bettale, J.-C. Faugére and L. Perret, Hybrid Approach for Solving Multivariate Systems over Finite Fields, Journal of Mathematical Cryptology, 3, pp. 177-197, 2009.
5. W. Beullens and B. Preneel, Field Lifting for Smaller UOV Public Keys, INDOCRYPT 2017, LNCS 10698, pp. 227-246, 2017.
6. W. Beullens, Improved Attacks on UOV and Rainbow, EUROCRYPT 2021, Part I, LNCS 12696, pp. 348-373, 2021.
7. W. Beullens, Breaking Rainbow Takes a Weekend on a Laptop, CRYPTO 2022, Part II, LNCS 13508, pp. 464-479, 2022.
8. A. Boldyreva, Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme, PKC 2003, LNCS 2567, pp. 31–46, 2003.
9. A. Bogdanov, T. Eisenbarth, A. Rupp and C. Wolf, Time-area Optimized Public-key Engines: MQ-cryptosystems as Replacement for Elliptic Curves?, CHES 2008, LNCS 5154, pp. 45-61, 2008.
10. D. Boneh, C. Gentry, B. Lynn, and H. Shacham, Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, EUROCRYPT 2003, LNCS 2656, pp. 416–432, 2003.
11. A.I.-T. Chen, M.S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E.L.-H. Kuo, F.Y.-S. Lee and B.-Y. Yang, SSE Implementation of Multivariate PKCs on Modern x86 CPUs, CHES'09, LNCS 5747, pp. 33-48, 2009.
12. P. Czypek, S. Heyse and E Thomae, Efficient Implementations of MQPKS on Constrained Devices, CHES 2012, LNCS 7428, pp. 374-389, 2012.
13. J. Ding, M-S. Chen, A. Petzoldt, D. Schmidt, and B-Y. Yang, Rainbow Technical report, National Institute of Standards and Technology, 2019. available at https://csrc.nist.gov/projects/ post-quantum-cryptography/round-2-submissions.
14. J. Ding, J. Deaton, Vishakha and Bo-Yin Yang, The Nested Subset Differential Attack: A Practical Direct Attack Against LUOV which Forges a Signature within 210 Minutes. IACR Cryptol. ePrint Arch. 2020: 967, 2020.
15. J. Ding and D. Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme, ACNS 2005, LNCS 3531, pp. 164-175, 2005.

16. H. Furue, K. Kinjo, Y. Ikematsu, Y. Wang, and T. Takagi, A Structural Attack on Block-Anti-Circulant UOV at SAC 2019, PQCrypto 2020, LNCS 12100, pp. 323–339, 2020.
17. H. Furue and M. Kudo, Polynomial XL: A Variant of the XL Algorithm Using Macaulay Matrices over Polynomial Rings, IACR Cryptol. ePrint Arch. 2021/1609, 2021.
18. Y. Hashimoto, On the security of Circulant UOV/Rainbow, IACR Cryptol. ePrint Arch. 2018/947, 2018.
19. Y. Hashimoto, On the security of Hufu-UOV, IACR Cryptol. ePrint Arch. 2021/1044, 2021.
20. A. Kipnis, J. Patarin, and L. Goubin, Unbalanced Oil and Vinegar Signature Schemes, CRYPTO'99, LNCS 1592, pp. 206-222, 1999.
21. T. Matsumoto, and H. Imai, Public Quadratic Polynomial-Tuples for efficient Signature-Verification and Message-Encryption, EUROCRYPT'88, LNCS 330, pp. 419-453, 1988.
22. A. Park, K. Shim, N. Koo, D. Han, Side-Channel Attacks on Post-Quantum Signature Schemes based on Multivariate Quadratic Equations: Rainbow and UOV, IACR Trans. Cryptogr. Hardw. Embed. Syst. 2018(3), pp. 500-523, 2018.
23. J. Patarin, Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms, EUROCRYPT'96, LNCS 1070, pp. 33-48, 1996.
24. Z. Peng and S. Tang, Circulant Rainbow: A New Rainbow Variant With Shorter Private Key and Faster Signature Generation, IEEE Access, vol. 5, pp. 11877 - 11886, 2017.
25. Z. Peng and S. Tang, Circulant UOV: a new UOV variant with shorter private key and faster signature generation, KSII Transactions on Internet and Information Systems (TIIS), vol. 12(3), pp. 1376-1395, 2018.
26. A. Petzoldt, S. Bulygin, and J. Buchmann, CyclicRainbow: A multivariate signature scheme with a partially cyclic public key, Indocrypt 2010, pp 33-48, 2010.
27. A. Petzoldt, M-S Chen, B-Y Yang, C. Tao and J. Ding, Design Principles for HFEv- Based Multivariate Signature Schemes, ASIACRYPT 2015, Part I, LNCS 9452, pp. 311-334, 2015.
28. Post-Quantum Cryptography, Round 2 Submissions, NIST Computer Security Resource Center, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions.
29. Post-Quantum Cryptography, Round 3 Submissions, NIST Computer Security Resource Center, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-3-Submissions.
30. K. Sakumoto, T. Shirai, H. Hiwatari: On Provable Security of UOV and HFE Signature Schemes against Chosen-Message Attack. PQCrypto 2011, LNCS vol. 7071, pp 68 - 82. Springer, 2011.
31. K-A. Shim and N. Koo, Algebraic Fault Analysis of UOV and Rainbow With the Leakage of Random Vinegar Values, IEEE Trans. Inf. Forensics Secur, 15, pp. 2429-2439, 2020.
32. K-A. Shim, S. Lee, N. Koo, Efficient Implementations of Rainbow and UOV using AVX2, IACR Trans. Cryptogr. Hardw. Embed. Syst. 2022(1), pp. 245-269, 2022.
33. D. Smith-Tone and R. Perlner, Rainbow band separation is better than we thought, IACR Cryptol. ePrint Arch. 2020/702, 2020.
34. A. Szepieniec and B. Preneel, Block-anti-circulant unbalanced oil and vinegar, SAC 2019, LNCS 11959, pp. 574–588, 2020.

35. C. Tao, A Method to Reduce the Key Size of UOV Signature Scheme, IACR Cryptol. ePrint Arch. 2019/473, 2-19.
36. C. Tao, A. Petzoldt and J. Ding, Efficient Key Recovery for All HFE Signature Variants, CRYPTO 2021 (I), pp. 70–93, 2021.
37. E. Thomae, About the Security of Multivariate Quadratic Public Key Schemes, Dissertation Thesis by Dipl. math. E. Thomae, RUB, 2013.
38. J. A. Verbel, J. Baena, D. Cabarcas, R. A. Perlner, and D. Smith-Tone, On the complexity of "superdetermined" minrank instances, PQCrypto 2019, LNCS 11505, pp. 167–186, 2019.
39. C. Wolf and B. Preneel, Large Superfluous Keys in Multivariate Quadratic Asymmetric Systems, PKC 2005, LNCS 3386, pp. 275-287, 2005.
40. B.-Y. Yang and J.-M. Chen, TTS: Rank Attacks in Tame-Like Multivariate PKCs. IACR Cryptology ePrint Archive, Report 2004/061, 2004. http://eprint.iacr.org/2004/061
41. B.-Y. Yang and J.-M. Chen, Building Secure Tame-like Multivariate Public-Key Cryptosystems: The new TTS, ACISP 2005, LNCS 3574, pp. 518-531, 2005.