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

Kyung-Ah Shim[1], Jeongsu Kim[1], and Youngjoo An[1]

National Institute for Mathematical Sciences
kashim,jsk2357,hellojoo@nims.re.kr

Abstract. Multivariate quadratic equations (MQ)-based public-key cryptographic algorithms are one of promising post-quantum replacements for current used public-key cryptography. After selecting to NIST Post-Quantum Cryptography Standardization Round 3 as one of digital signature finalists, Rainbow was cryptanalyzed by advanced algebraic attacks due to its multiple layered structure. In this document, we propose a new MQ-signature scheme based on UOV with a single layer that eliminates the causes of potential threats due to the multiple-layered structure. Our scheme uses sparse polynomials and the block inversion method using half-sized block matrices to reduce the secret key size and improve signing performance, respectively. We then provide security analysis, suggest secure parameters at three security categories and investigate their performance.

Keywords: Block matrix inversion · Multivariate quadratic equation · Sparse polynomial · 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 [20]. The MQ-based signature scheme with multiple layers such as Rainbow additionally requires the hardness of the MinRank problem.

Since Imai and Matsumoto [18] introduced the first MQ-encryption scheme, most of the MQ-schemes have been broken due to the structure related to the EIP problem except Unbalanced Oil-and-Vinegar (UOV) variants [17]. Rainbow, a variant of UOV, based on the multiple-layered structure to reduce the key size and improve performance [12]. It was selected as one of digital signature finalists in NIST PQC Standardization Round 3 [26]. Another MQ-signature scheme,

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

GeMMS, is one of the alternative candidates of NIST PQC Round 3. GeMMS is a special instance of HFEv- [24] which is a combination of the Minus and the Vinegar modifications with HFE in [20]. Recently, several advanced attacks on Rainbow [35, 1, 2, 30, 5, 6] including the MinRank attack and the RBS attacks and key recovery attack on HFEv- [33] have been proposed. Most of these advanced attacks on Rainbow are due to its multi-layered structure. The new attacks on Rainbow recovered the secret key at the security level 1 parameter of the NIST second-round submission in 53 hours on a laptop [6]. The author in [6] claimed that there seems to be some room for improvement for the attacks. Rainbow team proposed to NIST to replace the security level 1 (resp., 3) parameter with its security level 3 (resp., 5) parameter [10]. This change results in increasing the sizes of signatures and public keys. These attacks on Rainbow made UOV with a single layer a better choice both in terms of security and efficiency.

Although NIST recommended two algorithms, CRYSTALS-Dilithium and Falcon, as digital signature schemes in PQC Standardization Round 4. NIST announced a plan to issue a new Call for Proposals for signature schemes. The new proposal focuses on signature schemes that are not based on the structured lattices have short signatures and fast verification to diversify its signature portfolio. The MQ-signature scheme with a single layer such as UOV is emerging as a strong candidate. In this document, we propose a new efficient MQ-signature scheme based on the UOV structure with shorter secret key size and faster signing performance.

1.1 Design rationale, Advantages and Limitations

Design rationale, advantages and limitations of our scheme are as follows:

Simple Structure that Eliminates the Causes of Potential Threats. Rainbow using multiple layers is a variant of UOV to improve performance and reduce key sizes. However, its multiple-layered structure additionally requires the hardness of the MinRank problem and causes the recent advanced attacks. The MQ-signature scheme with a single layer, 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 and none of the attacks in [6] seem to apply to UOV [6]. The theoretical complexities of the known attacks against UOV are well-established. Our scheme maintains the structure of UOV which has long been believed to be secure and provides shorter secret key size and faster performance.

The Shortest Signature Length and Shorter Key Sizes. Signature length of our scheme is the shortest among post-quantum signature schemes based on the other hard problems. More precisely, it requires 134 bytes, 200 bytes and 260 bytes at security categories I, II, and V, respectively. Despite the shortest signature size and fast performance, the MQ-schemes suffer from relatively large key public/secret key sizes. Our scheme provides shorter secret key size than UOV by using sparse polynomials.

Fast Performance. In Rainbow with two layers, the number of equations is divided into two resulting in a reduction in the size of the matrix being inverted. However, the MQ-scheme with a single layer 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 [29] that exploits the inversions of half-sized matrices. Our scheme provide fast signing and verification performance.

Easy to Implement. Our scheme is very simple and is easy to understand and implement requiring basic linear algebra. More precisely, it requires simple operations such as matrix-vector products and solving linear systems over small finite fields without multi-precision arithmetic. It can be efficiently implemented on low cost devices without the need of a cryptographic coprocessor [7–9].

Protection Side-Channel Attacks. For resistance against side-channel attacks, UOV is secure against the correlation power analysis (CPA) presented in [19] by just using random affine maps instead of the equivalent keys without requiring an additional countermeasure. Thus, our scheme based on the UOV structure is also easy to design countermeasures against the CPA attacks. All key dependent operations in our scheme are performed in a time-constant manner

Additional Performance Improvements. There are additional improvements for signing and key generation by using precomputations and multi-cores, respectively.

- To speed up the signing process even more, we choose to split the signature generation in an offline and online phase, where the offline phase can already be performed before the message to be signed is known. Our scheme with precomputation is 15x to 60x faster than the original version without precomputation at the three security categories.
- Despite fast signing and verification performance, the key generation of our scheme is inefficient. To speed up key generation, we exploit multiple cores for independent operations resulting in 2x to 3x faster than the performance on a single core.

1.2 Related Works

There are several proposals for key size reductions of UOV or Rainbow. They use sparse polynomials (TTS, enTTS [37, 38] as special cases of Rainbow), random seeds for the part of public key or the whole secret key (CylicUOV, CyclicRainbow [23], CompressedRainbow [26]), the coefficients of the secret key in the smallest subfield $\mathbb{F}_2$ (so the coefficients of public key in $\mathbb{F}_2$, Lifted UOV (LUOV) [4, 25]) and circulant or toplitz matrices (Circulant-UOV [22] and Circulant-Rainbow [21], Block-anti-circulant UOV (BAC-UOV) [31], Hufu-UOV [32]) for the secret/public key reduction. However, in these schemes, reducing the secret key increases the public key size, reducing the public key increases the signature size or some reductions lead to heavy computational costs for signing and verification.

Most of these MQ-signature schemes were cryptanalyzed. Circulant-UOV [22] and Circulant-Rainbow [21] using circulant matrices to reduce the secret keys and speed up signing performance were entirely broken by Kipnis-Shamir attacks in [15]. LUOV was broken by a direct attack which forges a signature within 210 minutes [11]. Block-anti-circulant UOV (BAC-UOV) [31] using the block-anti-circulant matrices to reduce the public key was cryptanalyzed by Furue *et al.* [13]. The structural attacks on BAC-UOV [13] reduced the bit complexity up to 20% compared with the previously known attacks. Hashimoto [16] presented several algebraic attacks on Hufu-UOV, whose public key is generated by circulant matrices and toepliz matrices. Since some parts of the public key or the entire public key have the forms of structured matrices such as circulant matrices in BAC-UOV and Hufu-UOV, their public keys are distinguishable from random systems. It is contradict to make the public keys of the MQ-schemes to be hardly distinguishable from random systems. Consequently, to the best of our knowledge, there were no successful key size reduction results for UOV maintaining both the public key size and fast performance.

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 in UOV, $m = o$
- n: the number of variables in the public key, $n = m + v$.

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 (Unbalanced Oil and Vinegar) [17]. 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

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

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. Then a secret key is $(\mathcal{F}, T)$.

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

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

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 $\mathcal{F}_{L,C}^{(k)}$ are the part of linear terms and constant terms for $k = 1, \cdots, o$. In UOV, the central polynomial (1) can be written by

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

2.2 The Selection of Central Maps.

Now, we propose several types of the secret key depending on the selection of the quadratic parts, $\mathcal{F}_V^{(i)}$ and $\mathcal{F}_{OV}^{(i)}$, to achieve secret key reduction and faster signing performance.

- [**Selection of $\mathcal{F}_V^{(k)}$ using Sparse Polynomials.**] For the Vinegar × Vinegar quadratic parts, $\mathcal{F}_V^{(k)}$ for $k = 1, \cdots, o$,

$$\mathcal{F}_V^{(k)} = \mathcal{F}_{V,S}^{(k)} = \sum_{i=1}^{v} \alpha_i^k x_i x_{(i+k-1(mod\ v))+1},$$

where $\alpha_i^k \in_R \mathbb{F}_q^*$ $(i = 1, \cdots, v)$ so that the symmetric matrix of the quadratic part of $\mathcal{F}_V^{(k)}$ has full rank and all the quadratic terms in each $\mathcal{F}_V^{(k)}$ don't overlap for $k = 1, \cdots, o$.

- **Secret Key Size Reduction.** This technique reduces the size of the quadratic part $\mathcal{F}_V^{(k)}$ from $\frac{v \times v}{2} \cdot o$ field elements to $v \times o$ field elements.

– [**Selection of $\mathcal{F}_{OV}^{(k)}$ using Sparse Polynomials.**] For the Vinegar × Oil quadratic parts,

$$\mathcal{F}_{OV}^{(k)} = \mathcal{F}_{OV,S}^{(k)} = \sum_{i=1}^{o} \beta_i^k x_{v+i} x_{(i+k-1(mod\ o))+v+1},$$

where $\beta_i^k \in_R \mathbb{F}_q$ $(i = 1, \cdots, o)$ so that the symmetric matrix of the quadratic part of $\mathcal{F}_{OV}^{(k)}$ has full rank and all the quadratic terms in each $\mathcal{F}_{OV}^{(k)}$ don't overlap for $k = 1, \cdots, o$.

- **Secret Key Size Reduction.** This technique reduces the size of the quadratic part $\mathcal{F}_{OV}^{(k)}$ from $(v \times o) \cdot o$ field elements to $v \times o$ field elements.

According to the selections of $\mathcal{F}_V^{(k)}$ and $\mathcal{F}_{OV}^{(k)}$ $(k = 1, \cdots, o)$, there are several combinations for central maps:

– Sparse Vinegar ∗ Vinegar + Sparse Vinegar ∗ Oil:

$$\mathcal{F}_{SS}^{(k)} = \mathcal{F}_{V,S}^{(k)} + \mathcal{F}_{OV,S}^{(k)} + \mathcal{F}_{L,C}^{(k)}.$$

– Random Vinegar ∗ Vinegar + Sparse Vinegar ∗ Oil:

$$\mathcal{F}_{RS}^{(k)} = \mathcal{F}_{V,R}^{(k)} + \mathcal{F}_{OV,S}^{(k)} + \mathcal{F}_{L,C}^{(k)}.$$

– Sparse Vinegar ∗ Vinegar + Random Vinegar ∗ Oil:

$$\mathcal{F}_{SR}^{(k)} = \mathcal{F}_{V,S}^{(k)} + \mathcal{F}_{OV,R}^{(k)} + \mathcal{F}_{L,C}^{(k)}.$$

– Random Vinegar ∗ Vinegar + Random Vinegar ∗ Oil:

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

We describe our scheme based on the combinations of the above techniques. Our scheme provides the four types of secret keys using the sparse polynomials and random polynomials. The selection of random Vinegar ∗ Vinegar and random Vinegar ∗ Oil quadratic parts is the same as the key generation of UOV. Despite the different key generation methods, they share the same signing algorithm and verification algorithm. We use a random salt in signing for provable security [27].

■ **MQ-Sign**

- **KeyGen**(1^λ)**.** For a security parameter λ, choose a random invertible affine map $\mathcal{T}$ and do the followings:
    - **Secret Central Polynomials.** Choose $\mathcal{F}^{(k)}$ as follows:
        - ∗ MQ-Sign-SS: $\mathcal{F}^{(k)}_{SS} = \mathcal{F}^{(k)}_{V,S} + \mathcal{F}^{(k)}_{OV,S} + \mathcal{F}^{(k)}_{L,C}$.
        - ∗ MQ-Sign-RS: $\mathcal{F}^{(k)}_{RS} = \mathcal{F}^{(k)}_{V,R} + \mathcal{F}^{(k)}_{OV,S} + \mathcal{F}^{(k)}_{L,C}$.
        - ∗ MQ-Sign-SR: $\mathcal{F}^{(k)}_{SR} = \mathcal{F}^{(k)}_{V,S} + \mathcal{F}^{(k)}_{OV,R} + \mathcal{F}^{(k)}_{L,C}$.
        - ∗ MQ-Sign-RR: $\mathcal{F}^{(k)}_{RR} = \mathcal{F}^{(k)}_{V,R} + \mathcal{F}^{(k)}_{OV,R} + \mathcal{F}^{(k)}_{L,C}$.
    - Output a public key as $PK = \mathcal{P} = \mathcal{F} \circ \mathcal{T}$ and a secret key as $SK = (\mathcal{F}, \mathcal{T})$.
- **Sign**(SK, λ, M)**.** Given a message M and a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^o$, do the followings:
    - Choose a λ-bit random salt r, compute $\mathbf{h} = \mathcal{H}(M||r) \in \mathbb{F}_q^o$.
    - Compute $\mathbf{a} = \mathcal{F}^{-1}(\mathbf{h})$, i.e. $\mathcal{F}(\mathbf{a}) = \mathbf{h}$ as follows:
        - ∗ Select Vinegar values $s_V = (s_1, \cdots, s_v) \in \mathbb{F}_q^v$ at random and obtain a linear system of o equations with o unknowns $x_{v+1}, \cdots, x_{v+o}$ by substituting s_V into o central polynomials $\mathcal{F}^{(k)}$ for $1 \leq k \leq o$. After that, find a solution $(s_{v+1}, \cdots, s_{v+o})$ of the linear system using the BMI method.
        - ∗ If the linear systems is not solvable, choose another vector of Vinegar values s'_V and try again.
    - Compute $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{a})$ and 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 $\mathcal{P}$, check the equality $\mathcal{P}(\mathbf{z}) = \mathcal{H}(M||r)$. If the equality holds, output *valid.*

2.3 Solving Linear Systems

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

- **Substitution of Vinegar Values into the Central Polynomials.** Calculations for substituting random Vinegar values into the central polynomials are required. Since there are a large number of quadratic terms with Vinegar×Vinegar indexes and Vinegar×Oil indexes being substituted by the Vinegar values, the computations are heavy.
- **Solving Linear System.** Solving the linear systems after the Vinegar value substitution are required. 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 [29] that exploits the inversions of half-sized matrices.

Block Matrix Inversion Method. In signing, UOV and Rainbow use Gaussian elimination to solve the resulting linear system. In Rainbow implementation [26], the signing algorithm computes R^{-1} by using Gaussian elimination, where R is the coefficient matrix of the resulting liner system obtained from substituting the Vinegar values. In the single-layered scheme, the size of a matrix being inverted is one of the reasons for heavy computation. We use a fast method, the block matrix inversion (BMI) method proposed in [29], to reduce the size of a matrix being inverted for solving the linear system. The BMI method computes $R^{-1} \cdot \alpha$ directly, without finding R^{-1}: for a nonsingular 2×2 block matrix R, $R^{-1} \cdot \alpha$ requires two inversions, two matrix multiplications of the half-sized block matrices and four block matrix-vector products, where o is even. We describe the BMI method.

- **The 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

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

  Thus, $R^{-1} \cdot \alpha$ can be expressed by A^{-1} and the inverse of the Schur complement of A, $[D - CA^{-1}B]^{-1}$, if they exist,

$$R^{-1} \cdot \begin{pmatrix} \alpha_1 \\ \cdots \\ \alpha_o \end{pmatrix} = \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix} \begin{pmatrix} A^{-1} & O \\ 0 & [D - CA^{-1}B]^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -CA^{-1} & I \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \cdots \\ \alpha_o \end{pmatrix}.$$

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

$$A^{-1} \cdot \underline{\alpha} = \underline{\beta},\ C \cdot \underline{\beta},$$

$$[D - CA^{-1}B]^{-1} \cdot \underline{\gamma},\ (A^{-1}B) \cdot \underline{\gamma'}.$$

- **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 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 [29], 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.

2.4 Algorithm Specification

We describe KeyGen, Sign and Verify algorithms of our schemes in Algorithm 1, Algorithm 2, and Algorithm 3, respectively. In the Sign algorithm, A_{Sc} denotes the Schur complement of A, $[D - CA^{-1}B]^{-1}$.

Algorithm 1 KeyGen(λ)

```
**Require:** parameters $(q, v, o)$, length of salt $l$.
**Ensure:** key pair $(sk, pk)$.
$m \leftarrow o$
$n \leftarrow m + v$
**repeat**
    $M_T \leftarrow \texttt{Matrix}(q, n, n)$
**until** IsInvertible$(M_T)$ == **TRUE**
$\mathcal{T} \leftarrow M_T$
$Inv\mathcal{T} \leftarrow M_T^{-1}$
$\mathcal{F} \leftarrow \texttt{MQmap}(q, v, o)$
$\mathcal{P} \leftarrow \mathcal{F} \circ \mathcal{T}$
$sk \leftarrow (\mathcal{F}, Inv\mathcal{T}, l)$
$pk \leftarrow (\mathcal{P}, l)$
**Return** $(sk, pk)$
```

Algorithm 2 Sign(sk, M)

```
**Require:** message $M$, private key $(\mathcal{F},\ Inv\mathcal{T})$, length of the salt $l$.
**Ensure:** signature $\sigma = (\mathbf{z}, r) \in \mathbb{F}_q^n \times \{0,1\}^l$ such that $\mathcal{P}(\mathbf{z}) = \mathcal{H}(M||r)$.
**repeat**
    $y_1, ..., y_v \leftarrow_R \mathbb{F}_q$
    $\hat{f}^{(v_1+1)}, ..., \widehat{f^{(n)}} \leftarrow f^{(v+1)}(y_1, ..., y_v), ..., f^{(n)}(y, ..., y_v)$
    $(A, A_{Sc}, B, C, D, C_V) \leftarrow \texttt{Aff}^{-1}(\hat{f}^{(v+1)}, ..., \hat{f}^{(n)})$
**until** IsInvertible$(A, A_{Sc})$ == **TRUE**
$InvR = (A^{-1}, A_{Sc}^{-1})$
$r \leftarrow \{0,1\}^l$
$\mathbf{x} \leftarrow \mathcal{H}(M||r)$
$(y_{v+1}, ..., y_n) \leftarrow BMI(R, \mathcal{H}(M||r) - C_V)$
$\mathbf{z} = Inv\mathcal{T} \cdot \mathbf{y}$
$\sigma \leftarrow (\mathbf{z}, r)$
**Return** $\sigma$
```

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

2.5 Additional Improvements

There are additional performance improvements for signing and key generation by using precomputations and multi-cores, respectively.

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

[Offline Signing]

- After choosing random Vinegar values $s_V = (s_1, \cdots, s_v) \in \mathbb{F}_q^v$, substitute s_V into o equations $\mathcal{F}^{(k)}$ $(1 \leq k \leq o)$ to get the linear system R of o equations and o unknowns and a constant vector $c_V = (c_1, \cdots, c_o)$, where c_V is a vector of constant terms of $(\mathcal{F}^{(1)}(s_V), \cdots, \mathcal{F}^{(o)}(s_V))$.
- Compute A^{-1}, $A^{-1}B$, $C(A^{-1}B)$ and $[D - CA^{-1}B]^{-1}$.
- Store $< s_V, c_V, A^{-1}, A^{-1}B, C(A^{-1}B), [D - CA^{-1}B]^{-1} >$ as the precomputed values.

[Online Signing]

- Choose a random salt r and compute $h = \mathcal{H}(M||r)$ for a message M.
- Compute $R^{-1} \cdot h_V^{\mathsf{T}} = \alpha$ by using the precomputed values, where $h_V = (h_1 - c_1, \cdots, h_o - c_o)$ and $h = (h_1, \cdots, h_o)$.
- Compute $T^{-1} \cdot (S_V, \alpha)^{\mathsf{T}} = \sigma$ and output $\tau = (\sigma, r)$ as a signature on d.

Our scheme with precomputation is 10x to 50x faster than the original version without precomputation at the three security categories. According to the security analysis in [29], if some precomputed values together with signatures generated by them are exposed or reused then the secret key of our scheme is completely recovered. Thus, the precomputed values (actually, s_V) should be stored securely and should not be reused in signing.

Parallel Computation. Despite fast signing and verification performance, the key generation of our scheme is very inefficient. To speed up key generation, we exploit 10 cores for independent operations resulting in 2x to 3x faster than the performance on a single core.

3 Security Analysis

Now, we provide security analysis of our scheme and cost estimates against known algebraic attacks. We the suggest secure parameters at the three security categories. 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.

3.1 Hard Problems

The underlying problems are defined as follows:

- **MQ-Problem:** Given a system $\mathcal{P} = (P^{(1)}, \cdots, P^{(m)})$ of m qudratic 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 $(x'_1, \cdots, x'_n) \in \mathbb{F}_q^n$ such that $P^{(1)}(x'_1, \cdots, x'_n) = y_1, \cdots, P^{(m)}(x'_1, \cdots, x'_n) = y_m$.
- **EIP (Extended Isomorphism of Polynomials) 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 $\mathcal{F}' \in \mathcal{C}$.

3.2 Existential Unforgeability

In [27], 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 signaing algorithm to provide uniform distribution of the signatures. Their slightly modified UOV scheme is to use a random salt r as $\mathcal{H}(M||r)$ instead of $\mathcal{H}(M)$. Then the modified signature has the form $\tau = (\sigma, r)$, where σ is an original UOV. The existential unforgeability of our scheme follows the security proof of the modified UOV in [27].

3.3 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 known algebraic attacks: direct attacks, Kipnis-Shamir attacks, key recovery attacks using good keys and intersection attacks.

[**Direct Attacks.**] The most straightforward way to cryptanalyze the MQ-signature schemes is to solve the public system $\mathcal{P}(x) = \mathcal{H}(M||r)$. 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 [3, 14]. 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 categories 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 categories I, III, and V, respectively.

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 1. Complexity Estimates against Direct Attacks at the Security Category I.

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 2. Complexity Estimates against Direct Attacks at the Security Category III.

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

Table 3. Complexity Estimates against Direct Attacks at the Security Category V.

The complexity of $o = 46$ using the Polynomial XL algorithm [14] is about 131.25. We suggest another conservative parameter at the security category I with $o = 48$ of complexity 139.19.

[Key Recovery Attacks using Gook 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 [36, 34]. Wolf and Preneel [36] introduced the notion of equivalent keys as a fundamental tool to analyze the security of the MQ-schemes. Later, Thomae [34] 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 $(\mathcal{F}', \mathcal{T}')$ 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 don't 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 ($v = o$) [17]. 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 $P^{(i)} = T^T \cdot F^{(i)} \cdot T$, 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 [5], 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.

Finally, we summarize complexities of our scheme against all 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.

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	$C_{MQ}(1,\ ok(k+1)/2 - k(k-1),\ vk - o(k-1)$

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

[Implementation Attacks.] For resistance against the correlation power analysis (CPA) presented in [19], a random affine map $\mathcal{T}$ should be used instead of the equivalent key. For secure implementations, the Vinegar values required in signing must not be revealed or reused. To prevent fault attacks related to the Vinegar values in [28], it needs to check if the designated parts of the Vinegar values and the coefficients of the central polynomials are zero or not.

3.4 Parameter Selection

Now, we suggest secure parameters at the three security categories 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 attacks in Table 5 and Table 6

Security level	I	III	V
(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

Table 5. Suggested Parameters and Complexities of MQ-Sign(q, o, v) against Known Attacks

Additionally, we choose another conservative parameter for the security category I in Table 6.

Security Category I Parameter	$(\mathbb{F}_{2^8}, 48, 76)$
Direct Attack (Polynomila XL)	138.19
Intersection Attack	180.48

Table 6. Complexity Estimates for Conservative Parameter at Security Category I.

Key sizes and signature lengths of our scheme are given in Table 7, where PK, SK and Sig. Size represent the sizes of public keys, secret keys and signa-

Scheme	Security Category	I	III	V
Parameter	$(\mathbb{F}_q, o, v)$	$(\mathbb{F}_{2^8}, 46, 72)$	$(\mathbb{F}_{2^8}, 72, 112)$	$(\mathbb{F}_{2^8}, 96, 148)$
MQ-Sign-SS	Sig. Size	134	200	260
	PK	$328,441$	$1,238,761$	$2,892,961$
	SK	$15,561$	$37,729$	$66,421$
MQ-Sign-RS	Sig. Size	134	200	260
	PK	$328,441$	$1,238,761$	$2,892,961$
	SK	$133,137$	$485,281$	$1,110,709$
MQ-Sign-SR	Sig. Size	134	200	260
	PK	$328,441$	$1,238,761$	$2,892,961$
	SK	$164,601$	$610,273$	$1,416,181$
MQ-Sign-RR	Sig. Size	134	200	260
	PK	$328,441$	$1,238,761$	$2,892,961$
	SK	$282,177$	$1,057,825$	$2,460,469$

Table 7. Key/Signature Sizes of Our Schemes in Bytes.

tures, respectively. Since SS, RS, SR and RR have the different key generation algorithms depending on the selection of $\mathcal{F}_V$ and $\mathcal{F}_{OV}$, they different secret key sizes. However, they share the share the same parameters which results in the same signature size and the same public key sizes.

4 Implementation Details

We provide implementations of our scheme based on codes submitted to NIST PQC Standardization Round 3 [26] on our target platform.

4.1 Implementation Specification

We describe implementation specifications of our schemes.

- **Target Platform.** The computer we have used is equipped with an Intel(R) Core(TM) i7-6700X CPU at the constant clock frequency of 3.40GHz running Ubuntu 20.04LTS.
- **Random Number Generation and Hashing.** We use AES_CTR_DRBG as the random number generator. We use SHA-2 as the underlying hash function. In the SHA-2 hash function family, we use SHA256, SHA384, and SHA512 with output lengths of 256, 384, and 512 bits, respectively.
- **Selection of Finite Fields.** We choose $\mathbb{F}_q = \mathbb{F}_{2^8}$ as the underlying finite fields.
- **Use of Random Salts.** We use a random salts $r \in \{0,1\}^l$ to achieve provable security as in [27] which should be used only once.
- **Use of Equivalent Key and Linear Maps.** Our implementations use a secret key $\mathcal{T}$ as an equivalent key of the form $\mathcal{T} = \begin{pmatrix} I & T' \\ 0 & I \end{pmatrix}$ and a linear

map. While the central polynomials in our schemes except MQ-Sign-RR have linear part $\mathcal{F}_{L,C}^{(k)}$ and the public key has linear terms and no constant terms, MQ-Sign-RR has no linear terms which leads to reductions in the public/secret key sizes.

- **Use of the BMI Method with depth 1.** To reduce the size of matrices for solving linear systems, we use the BMI method once by representing 2×2 block matrices. After substituting the Vinegar values into the secret polynomials, we get a linear system of o equations and o variables and represent its coefficient matrix R as
$$R = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.$$
After computing A^{-1}, $A^{-1}B$, $C(A^{-1}B)$ and $[D - CA^{-1}B]^{-1}$ via two inversions and two matrix multiplications of $o/2 \times o/2$ block matrices, all remaining computations are made by four block matrix-vector products as
$$A^{-1} \cdot \underline{\alpha} = \underline{\beta},\ C \cdot \underline{\beta},$$
$$[D - CA^{-1}B]^{-1} \cdot \underline{\gamma},\ (A^{-1}B) \cdot \underline{\gamma'}.$$
If A or $[D - CA^{-1}B]$ is not invertible, it has to choose new Vinegar values and goes back to the first step.
- **Constant-time Implementation.** As in Rainbow implementation [26], all key dependent operations are performed in a time-constant manner. Therefore, our implementation is immune against timing attacks.

4.2 Implementation Results

We provide reference implementations and optimized implementations using AVX2 of our scheme on the target platform.

- The results presented in Table 8 and 9 include the numbers of CPU cycles required by the key generation, signing and verification.
- Each result of signing and verification (resp. key generation) is an average of 100,000 (resp. 10,000) measurements using the C programming language with GNU GCC version 9.4.0 compiler. Hyperthreading and Turbo Boost are switched off.
- Since our scheme provides different selection of secret keys in KeyGen, Table 7 and Table 8 show different results for key generation and signing. However, they share the same verification algorithm and the same parameters which result in the same performance in verification.

Scheme	Security Category	I	III	V
MQ-Sign-SS	KeyGen.	71,776,432	270,212,460	678,570,779
	Sign	464,141	1,013,754	1,818,507
	Verify	1,243,091	3,125,277	5,545,017
MQ-Sign-RS	KeyGen.	75,439,328	285,318,117	709,862,234
	Sign	1,035,362	2,276,898	4,124,725
	Verify	1,243,091	3,125,277	5,545,017
MQ-Sign-SR	KeyGen.	76,237,178	288,902,825	717,203,934
	Sign	201,834	707,959	1,486,775
	Verify	1,243,091	3,125,277	5,545,017
MQ-Sign-RR	KeyGen.	79,864,302	302,322,971	755,934,235
	Sign	1,303,024	3,333,303	6,577,958
	Verify	1,243,091	3,125,277	5,545,017

Table 8. Reference Implementations of Our Scheme in CPU cycles.

Scheme	Security Category	I	III	V
MQ-Sign-SS	KeyGen.	6,046,385	25,506,351	62,972,759
	Sign	154,645	378,634	471,085
	Verify	71,267	232,377	401,412
MQ-Sign-RS	KeyGen.	9,026,556	38,892,327	95,016,980
	Sign	174,790	447,656	626,373
	Verify	71,267	232,377	401,412
MQ-Sign-SR	KeyGen.	10,222,889	43,634,459	104,441,512
	Sign	166,987	417,445	630,000
	Verify	71,267	232,377	401,412
MQ-Sign-RR	KeyGen.	13,493,778	56,071,342	138,481,524
	Sign	184,761	491,738	708,415
	Verify	71,267	232,377	401,412

Table 9. Optimized Implementations of Our Scheme using AVX2 in CPU cycles.

5 Conclusion

We propose a new MQ-signature scheme based on UOV with a single layer with shorter secret key size and faster signing performance. Our scheme uses the block inversion method using half-sized blockmatrices and sparse polynomials to improve signing performance and reduce the secret key size, respectively. It provides fast signing and verification performance. Signature length of our scheme is the shortest among post-quantum signature schemes based on the other hard problems.

References

1. 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.
2. 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.
3. 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.
4. W. Beullens and B. Preneel, Field Lifting for Smaller UOV Public Keys, INDOCRYPT 2017, LNCS 10698, pp. 227-246, 2017.
5. W. Beullens, Improved Attacks on UOV and Rainbow, EUROCRYPT 2021, Part I, LNCS 12696, pp. 348-373, 2021.
6. W. Beullens, Breaking Rainbow Takes a Weekend on a Laptop, CRYPTO 2022, Part II, LNCS 13508, pp. 464-479, 2022.
7. 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.
8. 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.
9. P. Czypek, S. Heyse and E Thomae, Efficient Implementations of MQPKS on Constrained Devices, CHES 2012, LNCS 7428, pp. 374-389, 2012.
10. 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.
11. 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.
12. J. Ding and D. Schmidt. Rainbow, a New Multivariable Polynomial Signature Scheme, ACNS 2005, LNCS 3531, pp. 164-175, 2005.
13. 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.
14. 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.
15. Y. Hashimoto, On the security of Circulant UOV/Rainbow, IACR Cryptol. ePrint Arch. 2018/947, 2018.
16. Y. Hashimoto, On the security of Hufu-UOV, IACR Cryptol. ePrint Arch. 2021/1044, 2021.
17. A. Kipnis, J. Patarin, and L. Goubin, Unbalanced Oil and Vinegar Signature Schemes, CRYPTO'99, LNCS 1592, pp. 206-222, 1999.
18. T. Matsumoto, and H. Imai, Public Quadratic Polynomial-Tuples for efficient Signature-Verification and Message-Encryption, EUROCRYPT'88, LNCS 330, pp. 419-453, 1988.

19. 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.
20. 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.
21. 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.
22. 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.
23. A. Petzoldt, S. Bulygin, and J. Buchmann, CyclicRainbow: A multivariate signature scheme with a partially cyclic public key, Indocrypt 2010, pp 33-48, 2010.
24. 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.
25. Post-Quantum Cryptography, Round 2 Submissions, NIST Computer Security Resource Center, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-2-Submissions.
26. Post-Quantum Cryptography, Round 3 Submissions, NIST Computer Security Resource Center, https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-3-Submissions.
27. 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.
28. 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.
29. 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.
30. D. Smith-Tone and R. Perlner, Rainbow band separation is better than we thought, IACR Cryptol. ePrint Arch. 2020/702, 2020.
31. A. Szepieniec and B. Preneel, Block-anti-circulant unbalanced oil and vinegar, SAC 2019, LNCS 11959, pp. 574–588, 2020.
32. C. Tao, A Method to Reduce the Key Size of UOV Signature Scheme, IACR Cryptol. ePrint Arch. 2019/473, 2-19.
33. C. Tao, A. Petzoldt and J. Ding, Efficient Key Recovery for All HFE Signature Variants, CRYPTO 2021 (I), pp. 70–93, 2021.
34. E. Thomae, About the Security of Multivariate Quadratic Public Key Schemes, Dissertation Thesis by Dipl. math. E. Thomae, RUB, 2013.
35. 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.
36. C. Wolf and B. Preneel, Large Superfluous Keys in Multivariate Quadratic Asymmetric Systems, PKC 2005, LNCS 3386, pp. 275-287, 2005.
37. 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
38. 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.