Name of Proposal:

QR-UOV

Principal Submitter:
Hiroki Furue

email: furue-hiroki261@g.ecc.u-tokyo.ac.jp
phone: +81 3 5841 6940
organization: The University of Tokyo
postal address: 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan

Auxiliary Submitters:
Yasuhiko Ikematsu, Fumitaka Hoshino, Tsuyoshi Takagi,
Kan Yasuda, Toshiyuki Miyazawa, Tsunekazu Saito,
Akira Nagai

Inventors: All listed submitters

Owners: All listed submitters

Backup Point of Contact:
Akira Nagai
email: akira.nagai.td@hco.ntt.co.jp
phone: +81 422 59 7378
organization: NIPPON TELEGRAPH AND TELEPHONE CORPORATION
postal address: 3-9-11, Midori-cho, Musashino-shi, Tokyo, 180-8585, Japan

June 1, 2023

Contents

1 Introduction

1.1 History

Currently used public key cryptosystems such as RSA and ECC can be broken in polynomial time using a quantum computer executing Shor's algorithm [Sho99]. Thus, there has been growing interest in post-quantum cryptography (PQC), which is secure against quantum computing attacks. Indeed, the U.S. National Institute for Standards and Technology (NIST) has initiated a PQC standardization project [NIS].

Multivariate public key cryptography (MPKC), based on the difficulty of solving a system of multivariate quadratic polynomial equations over a finite field (the multivariate quadratic ($\mathcal{MQ}$) problem), is regarded as a strong candidate for PQC. The $\mathcal{MQ}$ problem is NP-complete [GJ90] and is thus likely to be secure in the post-quantum era.

The unbalanced oil and vinegar signature scheme (UOV) [KPG99], a multivariate signature scheme proposed by Kipnis et al. at EUROCRYPT 1999, has withstood various types of attacks for approximately 20 years. UOV is a well-established signature scheme owing to its short signature and short execution time. Indeed, a multilayer UOV variant Rainbow [DS05] was selected as a third-round finalist in the NIST PQC project [AASA$^+$19]. However, a new attack on Rainbow proposed by Beullens at 2022 [Beu22] broke the security of third round parameters and make the Rainbow scheme inefficient. Thus, the research following the approach to return to the original UOV has been accelerating. One problem of UOV is that the public key is much larger than those of other PQC candidates, for example, lattice-based signature schemes. Indeed, Rainbow, whose public key size is close to that of the plain UOV, had the largest public key among the third-round-finalist signature schemes, and NIST's report [AASA$^+$19] stated that Rainbow is unsuitable as a general-purpose signature scheme owing to this problem.

One of the approaches to solving this problem of the large public key of UOV is utilizing an algebraic structure. The CRYSTALS-DILITHIUM [DKL$^+$18] lattice-based signature scheme is one of the selected algorithms in the NIST PQC project. It is based on the hardness of the module learning with errors (MLWE) problem [BGV14]. As is well known, LWE [Reg09] is a confidential hard problem in cryptography, and the MLWE problem is a generalization of it using a module comprising vectors over a ring. This illustrates that a natural way to develop an efficient multivariate scheme with a small public key is to improve confidential schemes such as UOV in MPKC by investigating further algebraic theory. We present our QR-UOV following this direction to realize a UOV variant with a small public key.

1.2 Quotient Ring UOV at ASIACRYPT 2021

At ASIACRYPT 2021, Furue et al. [FIKT21] proposed a new variant of UOV, which is called *quotient ring UOV (QR-UOV)*. The public key of QR-UOV

is represented by block matrices in which every component corresponds to an element of a quotient ring $\mathbb{F}_q[x]/(f)$. More precisely, we use an injective ring homomorphism from the quotient ring $\mathbb{F}_q[x]/(f)$ to the matrix ring $\mathbb{F}_q^{\ell\times\ell}$, where $f \in \mathbb{F}_q[x]$ is a polynomial with $\deg f = \ell$. In this study, the image Φ_g^f of the homomorphism for $g \in \mathbb{F}_q[x]/(f)$ is called the *polynomial matrix* of g. From this homomorphism, we can compress the ℓ^2 components in Φ_g^f to ℓ elements of $\mathbb{F}_q$ because the polynomial matrix Φ_g^f is determined by the ℓ coefficients of g. This can be considered as a generalization of the block-anti-circulant UOV (BAC-UOV) presented at SAC 2019 [SP20], which is the case for $f = x^\ell - 1$. Utilizing the elements of a quotient ring in block matrices is similar to the MLWE problem [BGV14] because the MLWE problem uses elements of a ring in vectors. Namely, we can consider that the research undertaken to obtain from UOV to QR-UOV (including BAC-UOV) corresponds to that obtained from LWE to MLWE. Therefore, as with the MLWE problem, this type of research deserves more attention than passing notice.

To construct the QR-UOV, we must consider the symmetry of the polynomial matrices Φ_g^f. In UOV, the public key $\mathcal{P} = (p_1, \ldots, p_m)$, which comprises quadratic polynomials p_i, is obtained by composing a central map $\mathcal{F} = (f_1, \ldots, f_m)$ and a linear map $\mathcal{S}$, that is, $\mathcal{P} = \mathcal{F} \circ \mathcal{S}$. Then, the corresponding matrices $P_1, \ldots, P_m$ of the public key $\mathcal{P}$ are given by $P_i = S^\top F_i S$, where $F_1, \ldots, F_m$, and S are matrices corresponding to $\mathcal{F}$ and $\mathcal{S}$, respectively. If we choose $F_1, \ldots, F_m$, and S as block matrices, where the components are polynomial matrices Φ_g^f, the polynomial matrices must be stable under the transpose operation, namely, $(\Phi_g^f)^\top = \Phi_{g'}^f$ for some g'. Otherwise, $P_1, \ldots, P_m$ are not block matrices of Φ_g^f, and we cannot reduce the public key size using them. Polynomial matrices Φ_g^f are generally unstable under the transpose operation; therefore, we cannot directly use polynomial matrices Φ_g^f to construct an efficient UOV variant. To solve this problem, we introduce the concept of an $\ell \times \ell$ invertible matrix W such that $W\Phi_g^f$ is symmetric for any $g \in \mathbb{F}_q[x]/(f)$; that is, $W\Phi_g^f$ is stable under the transpose operation. In Theorem 1, we prove that there exists such symmetric W for any quotient ring $\mathbb{F}_q[x]/(f)$. Therefore, from equations

$$(\Phi_{g_1}^f)^\top (W\Phi_{g_2}^f)\Phi_{g_1}^f = (W\Phi_{g_1}^f)^\top \Phi_{g_2}^f \Phi_{g_1}^f = W\Phi_{g_1 g_2 g_1}^f,$$

we can construct a UOV variant using the quotient ring $\mathbb{F}_q[x]/(f)$ by choosing $F_1, \ldots, F_m$ as block matrices using $W\Phi_g^f$ and S as a block matrix with Φ_g^f.

Moreover, we should consider how the choice of f affects the security of the QR-UOV. Indeed, Furue et al. [FKI+20] broke BAC-UOV by transforming its anti-circulant matrices into diagonal concatenations of two smaller matrices. This transformation is obtained from the decomposition $x^\ell - 1 = (x-1)(x^{\ell-1} + \cdots + 1)$. Therefore, we investigate the relationship between the irreducibility of the polynomial f used to generate the quotient ring $\mathbb{F}_q[x]/(f)$ and the existence of such a transformation for symmetric matrices $W\Phi_g^f$. In Theorem 2 herein, we show that if f is irreducible (*i.e.,* $\mathbb{F}_q[x]/(f)$ is a field), then there is no such

transformation for matrices $W\Phi_g^f$, indicating that such an f is resistant to Furue et al.'s structural attack [FKI+20].

1.3 Our Purpose of this Document

In this document, we present a multivariate polynomial based digital signature scheme QR-UOV. The basic structure of QR-UOV is based on the original scheme at ASIACRYPT 2021 [FIKT21]. Moreover, we adopt the following developments [HFI+23, FI23, FIH+23] presented at SCIS 2023:

- The EUF-CMA security proof in the QROM is given, and we modify the signature generation for this proof. (The proof is mainly based on the result by Kosuge and Xagawa [KX22].)
- We provide a variety of parameter sets.
- We offer more optimized implementation and analyze its performance.
- We give a new security analysis using the rectangular MinRank attack.

Organizations The rest of this document is organized as follows. Section 2 prepares some notations. Section 3 gives preliminaries on the description of our QR-UOV. Section 4 and Section 5 describe the details of our algorithm and implementation, respectively. Section 6 provides the results of our performance analysis. Section 7 gives our security statements. Section 8 explains considerable attacks on QR-UOV. Section 9 discusses advantages and limitations of QR-UOV.

Acknowledgements We are grateful for help from Makoto Yanagisawa and Atsuhito Nakase. We also acknowledge Noriki Mo for pointing out inconsistencies between the spec and implementation and Testutaro Kobayashi for correcting a typo. We would like to thank Rika Akiyama and Satoshi Nakamura for their assistance in preparing the intermediate values. We are grateful to Shuhei Nakamura for his useful technical comments.

2 Notations and Parameters

This section describes the notations and parameters used in this document. We also define more specific basic methods used later in the specification.

2.1 Notations

bit	one of the two symbols '0' or '1'.
bit string	an ordered sequence of bits.
octet	a bit string of length 8.
octet string	an ordered sequence of octets
$\|\|$	a concatenation operator for two bit strings or for two octet strings.
$\mathbb{F}_q$	finite field with q elements for a prime power q
$\lceil x \rceil$	for x a real number returns the smallest integer greater than or equal to x.
$\lfloor x \rfloor$	for x a real number returns the largest integer less than or equal to x.
$[n]$	for n a positive integer returns the set $\{1, \ldots, n\}$.
$a \xleftarrow{\$} \mathrm{A}$	$a \in \mathrm{A}$ is chosen uniformly at random from A.

We here also give some notations for representation matrices of elements of a quotient ring described in Subsection 3.2.

f	an irreducible polynomial in $\mathbb{F}_q[x]$ with degree ℓ
Φ_g^f	an $\ell \times \ell$ matrix over $\mathbb{F}_q$ defined by equation (4) in Subsection 3.2
$\mathcal{A}_f$	$\left\{ \Phi_g^f \in \mathbb{F}_q^{\ell \times \ell} \;\middle|\; g \in \mathbb{F}_q[x]/(f) \right\}$
W	an $\ell \times \ell$ matrix over $\mathbb{F}_q$ such that WX is symmetric for any $X \in \mathcal{A}_f$
$W\mathcal{A}_f$	$\{WX \in \mathbb{F}_q^{\ell \times \ell} \mid X \in \mathcal{A}_f\}$
$\mathcal{A}_f^{a,b}$	the set of $a\ell \times b\ell$ block matrices whose each component is an element of $\mathcal{A}_f$
$W^{(a)}$	the $a\ell \times a\ell$ block diagonal matrix concatenating W diagonally a times

2.2 Parameters

ℓ, V, M	positive integers
v	number of vinegar variables: $v = \ell \cdot V$
m	number of oil variables (equals to # of equations): $m = \ell \cdot M$
n	number of variables: $n = v + m$
N	$N = V + M$
λ	secuirty parameter
r	a random λ-bit string

2.3 MGF1

MGF1 is a mask generation function parameterized by a hash function. MGF1 is defined in [MKJR16] and is also called KDF1 in [Sho01].

3 Preliminaries

Our QR-UOV is an extension of the plain UOV, and thus we recall the construction of the plain UOV to describe QR-UOV smoothly in Section 4. Furthermore, as preliminaries for the construction of our QR-UOV, we introduce matrices representing elements of a quotient ring.

3.1 Basic Description of UOV used for QR-UOV

This subsection describes the structure of the unbalanced oil and vinegar signature scheme (UOV) [KPG99]. For variables $\mathbf{x} = (x_1, \ldots, x_n)$ over $\mathbb{F}_q$, we call $x_1, \ldots, x_v$ *vinegar variables* and $x_{v+1}, \ldots, x_n$ *oil variables*.

We first recall the key generation of UOV as follows: We design $\mathcal{F} = (f_1, \ldots, f_m) : \mathbb{F}_q^n \to \mathbb{F}_q^m$, called a *central map*, such that each f_k with $k \in [m]$ is a quadratic polynomial of the form

$$f_k(x_1, \ldots, x_n) = \sum_{i=1}^{v} \sum_{j=i}^{n} \alpha_{i,j}^{(k)} x_i x_j \tag{1}$$

where $\alpha_{i,j}^{(k)} \in \mathbb{F}_q$. Next, we choose a random linear map $\mathcal{S} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ to hide the structure of $\mathcal{F}$. The public key $\mathcal{P}$ is then provided as a polynomial map,

$$\mathcal{P} = \mathcal{F} \circ \mathcal{S} : \mathbb{F}_q^n \to \mathbb{F}_q^m, \tag{2}$$

and the secret key comprises $\mathcal{F}$ and $\mathcal{S}$. We here omit linear and constant terms of $\mathcal{F}$ and constant terms of $\mathcal{S}$ for simplicity.

Next, we describe the inversion of the central map $\mathcal{F}$. When we find $\mathbf{x} \in \mathbb{F}_q^n$ satisfying $\mathcal{F}(\mathbf{x}) = \mathbf{t}$ for a given $\mathbf{t} \in \mathbb{F}_q^m$, we first choose random values $y_1, \ldots, y_v$ in $\mathbb{F}_q$ as the values of the vinegar variables. We can then easily obtain a solution for the equation $\mathcal{F}(y_1, \ldots, y_v, x_{v+1}, \ldots, x_n) = \mathbf{t}$, because this is a linear system of m equations in m oil variables from the construction of the central map (1). If there is no solution to this equation, we choose new random values $y_1', \ldots, y_v'$, and repeat the above procedure.

By using this inversion approach, the signature is generated as follows: Given a message $\mathbf{m} \in \mathbb{F}_q^m$ to be signed, find a solution $\mathbf{m}_1$ to the equation $\mathcal{F}(\mathbf{x}) = \mathbf{m}$, and this gives a signature $\mathbf{s} = \mathcal{S}^{-1}(\mathbf{m}_1) \in \mathbb{F}_q^n$ for the message $\mathbf{m}$. The verification is performed by confirming whether $\mathcal{P}(\mathbf{s}) = \mathbf{m}$.

Finally, we introduce matrices representing the public and secret keys of UOV. For each polynomial p_i of the public key $\mathcal{P}$, there exists an $n \times n$ matrix P_i such that $p_i(\mathbf{x}) = \mathbf{x}^\top \cdot P_i \cdot \mathbf{x}$. Similarly, an $n \times n$ matrix F_i can be taken for each f_i with $1 \leq i \leq m$, and an $n \times n$ matrix S is defined to satisfy $\mathcal{S}(\mathbf{x}) = S \cdot \mathbf{x}$.

In general, these matrices P_i and F_i are taken as symmetric matrices if q is odd, and are taken as upper triangular matrices if q is even. For these representation matrices, based on equation (1), F_i has the following form

$$\begin{pmatrix} *_{v\times v} & *_{v\times m} \\ *_{m\times v} & 0_{m\times m} \end{pmatrix}. \tag{3}$$

Furthermore, from $\mathcal{P} = \mathcal{F} \circ \mathcal{S}$, we have

$$P_i = S^\top F_i S, \ \ (i \in [m]).$$

3.2 Matrix Representation of Quotient Ring Elements

We here introduce polynomial matrices representing elements of a quotient ring.

Let ℓ be a positive integer and $f \in \mathbb{F}_q[x]$ with $\deg f = \ell$. For any element g of the quotient ring $\mathbb{F}_q[x]/(f)$, we can uniquely define an $\ell \times \ell$ matrix Φ_g^f over $\mathbb{F}_q$ such that

$$\begin{pmatrix} 1 & x & \cdots & x^{\ell-1} \end{pmatrix} \Phi_g^f = \begin{pmatrix} g & xg & \cdots & x^{\ell-1}g \end{pmatrix}. \tag{4}$$

From this equation, we have

$$x^{j-1}g = \sum_{i=1}^{\ell} \left(\Phi_g^f\right)_{ij} \cdot x^{i-1} \quad (1 \le j \le \ell),$$

and $\left(\Phi_g^f\right)_{ij}$ is the coefficient of x^{i-1} in $x^{j-1}g$. We call such a matrix Φ_g^f the *polynomial matrix* of g. The following lemma can be easily derived from this definition:

Lemma 1. *For any $g_1, g_2 \in \mathbb{F}_q[x]/(f)$, we have*

$$\Phi_{g_1}^f + \Phi_{g_2}^f = \Phi_{g_1+g_2}^f, \ \Phi_{g_1}^f \Phi_{g_2}^f = \Phi_{g_1 g_2}^f.$$

That is, the map $g \mapsto \Phi_g^f$ is an injective ring homomorphism from $\mathbb{F}_q[x]/(f)$ to the matrix ring $\mathbb{F}_q^{\ell\times\ell}$.

We let the algebra of the matrices $\mathcal{A}_f := \left\{\Phi_g^f \in \mathbb{F}_q^{\ell\times\ell} \;\middle|\; g \in \mathbb{F}_q[x]/(f)\right\}$. This is a subalgebra in the matrix algebra $\mathbb{F}_q^{\ell\times\ell}$ from Lemma 1. Every $\ell\times\ell$ polynomial matrix Φ_g^f in $\mathcal{A}_f$ can be represented by only ℓ elements in $\mathbb{F}_q$, because Φ_g^f is determined by the ℓ coefficients of $g \in \mathbb{F}_q[x]/(f)$. Our QR-UOV compresses the public key size of UOV by utilizing this property of Φ_g^f.

For the construction of QR-UOV, we also give the concept of a matrix $W \in \mathbb{F}_q^{\ell\times\ell}$ such that $W\Phi_g^f$ is stable under the transpose operation. Then, note that any matrix in $W\mathcal{A}_f := \{WX \in \mathbb{F}_q^{\ell\times\ell} \mid X \in \mathcal{A}_f\}$ can also be represented by only ℓ elements in $\mathbb{F}_q$. In the following theorem, we prove that there exists an invertible matrix W for any f.

Theorem 1 (Theorem 1 in [FIKT21]). *Let $f \in \mathbb{F}_q[x]$ with $\deg f = \ell$. Then, there exists an invertible matrix $W \in \mathbb{F}_q^{\ell\times\ell}$ such that WX is a symmetric matrix for any $X \in \mathcal{A}_f$.*

In the proof of Theorem 1 in [FIKT21], they propose a way of constructing such a W from a nonzero linear map $\phi : \mathbb{F}_q[x]/(f) \to \mathbb{F}_q$ such that the ij-component of W is equal to $\phi(x^{i+j-2})$.

For $\mathcal{A}_f$ and positive integers N, we define the the set $\mathcal{A}_f^{N,N}$ of block matrices in $\mathbb{F}_q^{\ell N\times\ell N}$ whose every component is an element of $\mathcal{A}_f$. These matrices are utilized for the construction of our QR-UOV in Section 4.

Example 1. *For $f = x^3 - 3x - 1$ in $\mathbb{F}_7[x]$, we can take one element of $A_f^{2,2}$ as follows*

$$\begin{pmatrix} 2 & 0 & 2 & 0 & 0 & 1 \\ 2 & 2 & 6 & 1 & 0 & 3 \\ 0 & 2 & 2 & 0 & 1 & 0 \\ 3 & 2 & 5 & 3 & 6 & 5 \\ 5 & 2 & 3 & 5 & 0 & 0 \\ 2 & 5 & 2 & 6 & 5 & 0 \end{pmatrix}.$$

Every 3×3 block of this matrix can be represented as an element of $\mathbb{F}_q[x]/(f)$, that is, this matrix can be represented as a 2×2 matrix over $\mathbb{F}_q[x]/(f)$

$$\begin{pmatrix} 2+2x & x \\ 3+5x+2x^2 & 3+5x+6x^2 \end{pmatrix}.$$

In the rest of this document, we construct QR-UOV using $\mathcal{A}_f$ with an irreducible f for the security of QR-UOV. See Subsection 8.1 for the reason that we use an irreducible polynomial f in QR-UOV.

4 Algorithm Specification

This section mainly provides the following

- the description of key generation, signature generation, and verification algorithms,
- proposed parameter sets,
- some remarks for our implementation.

We describe the QR-UOV scheme as proposed in [FIKT21] and add a modification for the security proof used in [FIH+23].

4.1 Key Generation

Let v be the number of vinegar variables, m be the number of oil variables which is equal to the number of equations, and $n = v + m$. From the notations

in Subsection 2.1, the public and secret keys of QR-UOV are represented by elements of $\mathcal{A}_f^{N,N}$ and $W^{(N)}\mathcal{A}_f^{N,N} := \left\{W^{(N)} \cdot X \,|X \in \mathcal{A}_f^{N,N}\right\}$, where $N = n/\ell$ with the number n of variables. Note that we here use an irreducible polynomial as f of $\mathcal{A}_f$ for the reason in Subsection 8.1. This subsection presents the key generation of our QR-UOV. See Algorithm 1 for more details.

The standard key generation of QR-UOV is described as follows:

1. Choose F_i $(i \in [m])$ from $W^{(N)}\mathcal{A}_f^{N,N}$ as a symmetric matrix with the lower-right $m \times m$ zero-block as in (3).
2. Choose an invertible matrix S from $\mathcal{A}_f^{N,N}$ randomly.
3. Compute the public key $P_i = S^\top F_i S$ $(i \in [m])$.

Then, P_i $(i \in [m])$ representing the public key map are elements of $W^{(N)}\mathcal{A}_f^{N,N}$ from the following proposition:

Proposition 1 (Prop. 1 in [FIKT21])**.** *For $X \in \mathcal{A}_f^{N,N}$ and $Y \in W^{(N)}\mathcal{A}_f^{N,N}$, we have*

$$X^\top Y X \in W^{(N)}\mathcal{A}_f^{N,N}.$$

Subsequently, we apply an improved method restricting the secret key $\mathcal{S}$ to a specific compact form, which was first proposed by Czypek et al. [CHT12]. Before describing the improved method, we prepare some notations: For the public key P_i $(i \in [m])$ and the secret key F_i $(i \in [m])$, we define submatrices as follows

$$P_i = \begin{pmatrix} P_{i,1} & P_{i,2} \\ P_{i,2}^\top & P_{i,3} \end{pmatrix},$$
$$F_i = \begin{pmatrix} F_{i,1} & F_{i,2} \\ F_{i,2}^\top & 0_{m\times m} \end{pmatrix},$$

where $P_{i,1}$ and $F_{i,1}$ are symmetric $v\times v$ matrices, $P_{i,2}$ and $F_{i,2}$ are $v\times m$ matrices, and $P_{i,3}$ is a symmetric $m \times m$ matrix. We then suppose to limit the secret key S to the following compact form

$$S = \begin{pmatrix} I_v & S' \\ O & I_m \end{pmatrix}, \tag{5}$$

where S' is a $v \times m$ matrix. Then, from $P_i = S^\top F_i S$ $(i \in [m])$, we obtain

$$\begin{aligned} F_{i,1} &= P_{i,1}, \\ F_{i,2} &= -P_{i,1}S' + P_{i,2}, \\ 0_{m\times m} &= {S'}^\top P_{i,1} S' - P_{i,2}^\top S' - {S'}^\top P_{i,2} + P_{i,3}. \end{aligned} \tag{6}$$

By using this equation, in the improved key generation step, $P_{i,1} \in W^{(V)}\mathcal{A}_f^{V,V}$, $P_{i,2} \in W^{(V)}\mathcal{A}_f^{V,M}$ $(i \in [m])$, and $S' \in \mathcal{A}_f^{V,M}$, where $V = v/\ell$ and $M = m/\ell$,

are first generated from random seeds, and $P_{i,3} \in W^{(M)}\mathcal{A}_f^{M,M}$ $(i \in [m])$ is computed by

$$P_{i,3} = -{S'}^{\top} P_{i,1} S' + P_{i,2}^{\top} S' + {S'}^{\top} P_{i,2}.$$

As a result, the public key is composed of $m \times m$ matrices $P_{i,3}$ $(i \in [m])$ and the 2λ-bit seed $\mathsf{seed}_{\mathsf{pk}}$ for $P_{i,1}$, $P_{i,2}$ $(i \in [m])$, and the secret key is composed of the 2λ-bit seed $\mathsf{seed}_{\mathsf{sk}}$ for S', where λ is the security parameter. The security of QR-UOV is not weakened by this optimization, since this does not affect the distribution of the public and secret keys.

We here use the following two functions $\mathsf{Expand}_{\mathsf{sk}}$ and $\mathsf{Expand}_{\mathsf{pk}}$ to expand the public and secret keys from randomly chosen seeds

$\mathsf{Expand}_{\mathsf{sk}}$ This expands the seed $\mathsf{seed}_{\mathsf{sk}}$ for the secret key to $S' \in \mathcal{A}_f^{V,M}$. As we mentioned before, this S' can be represented as a $V \times M$ matrix over $\mathbb{F}_q[x]/(f)$. We sample the matrix in row-major order and sample each polynomial in $\mathbb{F}_q[x]/(f)$ in reverse degree order from the constant term to the coefficient of $x^{\ell-1}$. See Subsection 4.6 for the hash function used to generate polynomials.

$\mathsf{Expand}_{\mathsf{pk}}$ This expands the seed $\mathsf{seed}_{\mathsf{pk}}$ for the public key to $\{P_{i,1}\}_{i\in[m]}, \{P_{i,2}\}_{i\in[m]}$ where $P_{i,1}$ is a symmetric $v \times v$ matrix in $W^{(V)}\mathcal{A}_f^{V,V}$ and $P_{i,2}$ is a $v \times m$ matrix in $W^{(V)}\mathcal{A}_f^{V,M}$. Then, this $P_{i,1}$ and $P_{i,2}$ can be represented as $V \times V$ and $V \times M$ matrices over $\mathbb{F}_q[x]/(f)$. We here first sample $P_{1,1}, \ldots, P_{m,1}$ and then $P_{1,2}, \ldots, P_{m,2}$. For each matrix, we sample in row-major order and sample each polynomial in $\mathbb{F}_q[x]/(f)$ in reverse degree order. Note that for $P_{i,1}$ we sample only the upper-triangular elements due to the symmetry.

Finally, we compare the public key size of the plain QR-UOV with that of the improved QR-UOV. The public key of the plain QR-UOV is represented by $P_{i,1}$, $P_{i,2}$, and $P_{i,3}$ $(i \in [m])$, and that of the improved QR-UOV uses a seed $\mathsf{seed}_{\mathsf{pk}}$ and $P_{i,3}$ $(i \in [m])$. Thus, the number of elements in $\mathbb{F}_q$ needed to represent the public key of the plain QR-UOV is

$$mn(n+\ell)/2\ell,$$

whereas that of the improved QR-UOV is

$$m^2(m+\ell)/2\ell.$$

4.2 Signature Generation

Our signature generation of QR-UOV is mainly depending on the standard signature generation of the plain UOV: Invert the central map $\mathcal{F}$ by fixing v values of the vinegar variables, and then multiply S^{-1} in the form of

$$S^{-1} = \begin{pmatrix} I_v & -S' \\ O & I_m \end{pmatrix},$$

Algorithm 1 KeyGen()

Input: parameters (q, v, m, ℓ), security parameter λ
Output: public key pk, secret key sk

1: $\mathsf{seed}_{\mathsf{pk}}, \mathsf{seed}_{\mathsf{sk}} \xleftarrow{\$} \{0,1\}^{2\lambda}$
2: $\{P_{i,1}\}_{i\in[m]}, \{P_{i,2}\}_{i\in[m]} \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{seed}_{\mathsf{pk}})$
$\triangleright\ P_{i,1} \in W^{(V)}\mathcal{A}_f^{V,V}$ (symmetric), $P_{i,2} \in W^{(V)}\mathcal{A}_f^{V,M}$
3: $S' \leftarrow \mathsf{Expand}_{\mathsf{sk}}(\mathsf{seed}_{\mathsf{sk}})$ $\triangleright\ S' \in A_f^{V,M}$
4: **for** i from 1 to m **do**
5: $\quad P_{i,3} \leftarrow -{S'}^{\top} P_{i,1} S' + P_{i,2}^{\top} S' + {S'}^{\top} P_{i,2}$
6: **end for**
7: **return** $(\mathsf{pk}, \mathsf{sk}) = \left(\left(\mathsf{seed}_{\mathsf{pk}}, \{P_{i,3}\}_{i\in[m]}\right), \mathsf{seed}_{\mathsf{sk}}\right)$

from equation (5). We here add a modification for the EUF-CMA security proof proposed by Sakumoto et al. [SSH11]. See Algorithm 2 for more details.

We here describe the inversion of the central map $\mathcal{F}$ in our modified signature generation. We first choose values for the vinegar variables $y_1, \ldots, y_v$ randomly. We then choose λ-bit random salt r and compute $\mathbf{t} \in \mathbb{F}_q^m$ by applying a hash function Hash on the input concatenating a given message $\mathbf{M}$ and the salt r, namely $\mathbf{t} := \mathsf{Hash}(\mathbf{M}||r)$. If the linear system for the oil variables $x_{v+1}, \ldots, x_n$

$$\mathcal{F}(y_1, \ldots, y_v, x_{v+1}, \ldots, x_n) = \mathbf{t}, \tag{7}$$

has solutions, then we obtain the signature by applying $\mathcal{S}^{-1}$ into $(y_1, \ldots, y_v, y_{v+1}, \ldots, y_n)$, where $(y_{v+1}, \ldots, y_n)$ is a randomly chosen solution of equation (7). If there exists no solution of equation (7), then we choose a new salt and update $\mathbf{t}$ until equation (7) has solutions.

The main difference from the standard signature generation algorithm is that if equation (7) has no solution, then we choose a new random salt instead of choosing new vinegar variables. By doing so, the signature $\mathbf{s}$ satisfying $\mathcal{P}(\mathbf{s}) = \mathsf{Hash}(\mathbf{M}||r)$ is uniformly distributed in $\mathbb{F}_q^n$, and this fact enables us to prove the EUF-CMA security of QR-UOV in Subsection 7.1. For the efficiency, we confirm that the expected number of computing $\mathbf{t} = \mathsf{Hash}(\mathbf{M}||r)$ until equation (7) has solutions is approximately 2.0 for any parameter sets by assuming that equation (7) is a randomized system for $x_{v+1}, \ldots, x_n$.

4.3 Signature Verification

Our signature verification of QR-UOV is the same as that of the plain UOV. Given the public key pk, a message $\mathbf{M}$, and a signature $\sigma = (r, \mathbf{s})$, the authenticity of the signature is checked as follows:

- Use the hash function Hash to compute $\mathbf{t} = \mathsf{Hash}(\mathbf{M}||r)$.

Algorithm 2 Sign($\mathbf{M}, \mathsf{pk}, \mathsf{sk}$)

Input: message $\mathbf{M}$, public key pk, secret key sk
Output: signature σ

$\left(\mathsf{seed}_{\mathsf{pk}}, \{P_{i,3}\}_{i\in[m]}\right) \leftarrow \mathsf{pk}$
$\mathsf{seed}_{\mathsf{sk}} \leftarrow \mathsf{sk}$
$\{P_{i,1}\}_{i\in[m]}, \{P_{i,2}\}_{i\in[m]} \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{seed}_{\mathsf{pk}})$
$S' \leftarrow \mathsf{Expand}_{\mathsf{sk}}(\mathsf{seed}_{\mathsf{sk}})$
for i from 1 to m **do**
  $F_{i,1} \leftarrow P_{i,1}$
  $F_{i,2} \leftarrow -P_{i,1}S' + P_{i,2}$
end for
$S \leftarrow \begin{pmatrix} I_v & S' \\ 0_{m\times v} & I_m \end{pmatrix}$
$\mathbf{y} = (y_1, \ldots, y_v)^\top \xleftarrow{\$} \mathbb{F}_q^v$
$L \leftarrow \begin{pmatrix} 2\mathbf{y}^\top F_{1,2} \\ \vdots \\ 2\mathbf{y}^\top F_{m,2} \end{pmatrix}$ ▷ $L \in \mathbb{F}_q^{m\times m}$
$\mathbf{u} \leftarrow \left(\mathbf{y}^\top F_{1,1}\mathbf{y}, \ldots, \mathbf{y}^\top F_{m,1}\mathbf{y}\right)^\top$ ▷ $\mathbf{u} \in \mathbb{F}_q^m$
repeat
  $r \xleftarrow{\$} \{0,1\}^\lambda$
  $\mathbf{t} \leftarrow \mathsf{Hash}(\mathbf{M}||r)$ ▷ $\mathbf{t} \in \mathbb{F}_q^m$
until $L\mathbf{x} = \mathbf{t} - \mathbf{u}$ has solutions for $\mathbf{x}$.
Choose one solution $(y_{v+1}, \ldots, y_n) \in \mathbb{F}_q^m$ of $L\mathbf{x} = \mathbf{t} - \mathbf{u}$ randomly.
$\mathbf{s} \leftarrow S^{-1}(y_1, \ldots, y_v, y_{v+1}, \ldots, y_n)^\top$
return $\sigma = (r, \mathbf{s})$

- Compute $\mathbf{t}' \in \mathbb{F}_q^m$ by substituting the signature $\mathbf{s} \in \mathbb{F}_q^n$ for the public key map $\mathcal{P}$ (namely $\mathbf{t}' = \mathcal{P}(\mathbf{s})$).

If $\mathbf{t} = \mathbf{t}'$ holds, the signature σ is accepted, otherwise it is rejected. See Algorithm 3 for more details.

4.4 Parameter Choice

We propose some parameter sets of QR-UOV. These parameter sets are proposed in accordance with security levels I, III, and V of the NIST PQC project. We take 7, 31, and 127 as the number q of the finite field. The reason that we do not use a finite field with even characteristics is as follows: If q is even, in a polynomial obtained as $\mathbf{x}A\mathbf{x}^\top$ where $A \in W^{(N)}\mathcal{A}_f^{N,N}$, the coefficients corresponding to the non-diagonal components of every diagonal block are zero owing to the symmetry of $W\Phi_g^f$. For each security level, we propose four parameter

Algorithm 3 Verify$(\mathbf{M}, \mathsf{pk}, \sigma)$

Input: message $\mathbf{M}$, public key pk, signature σ
Output: accept or reject

$\left(\mathsf{seed}_{\mathsf{pk}}, \{P_{i,3}\}_{i\in[m]}\right) \leftarrow \mathsf{pk}$
$(r, \mathbf{s}) \leftarrow \sigma$
$\{P_{i,1}\}_{i\in[m]}, \{P_{i,2}\}_{i\in[m]} \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{seed}_{\mathsf{pk}})$
for i from 1 to m **do**
  $P_i \leftarrow \begin{pmatrix} P_{i,1} & P_{i,2} \\ P_{i,2}^\top & P_{i,3} \end{pmatrix}$
end for
$\mathbf{t} \leftarrow \mathsf{Hash}(\mathbf{M}||r)$
$\mathbf{t}' \leftarrow \left(\mathbf{s}^\top P_1 \mathbf{s}, \ldots, \mathbf{s}^\top P_m \mathbf{s}\right)^\top$
return accept if $\mathbf{t} = \mathbf{t}'$ and reject otherwise.

sets. (See Table 1.) These parameters are determined based on q and ℓ for the following purposes

$q = 7, \ell = 10$: to realize a fast implementation,

$q = 31, \ell = 3$: to make the signature size small,

$q = 31, \ell = 10$: to make the public key size small,

$q = 127, \ell = 3$: to vary the order of the finite fields.

Furthermore, we here set the security parameter λ as 128, 192, and 256 for the security level I, III, and V, respectively.

In QR-UOV, any irreducible polynomial with degree ℓ over $\mathbb{F}_q$ can be taken as f. We here show one example of f and $W \in \mathbb{F}_q^{\ell\times\ell}$ used in our implementations for each set of q and ℓ.

$$q = 7, \ell = 10 \ : \ f = x^{10} - 2x - 1, \ W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$q = 31, \ell = 3 \ : \ f = x^3 - x - 1, \ W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

Table 1: Proposed parameters for security level I, III, and V

SL	q	v	m	ℓ
I	7	740	100	10
	31	165	60	3
	31	600	70	10
	127	156	54	3
III	7	1100	140	10
	31	246	87	3
	31	890	100	10
	127	228	78	3
V	7	1490	190	10
	31	324	114	3
	31	1120	120	10
	127	306	105	3

$$q = 31, \ell = 10 \ : \ f = x^{10} - 5x^3 - 1, W = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$q = 127, \ell = 3 \ : \ f = x^3 - x - 1, W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

4.5 Note on the Hash Function

Basically, we recommend using the hash functions SHAKE128 or SHAKE256, which can output an arbitrary-length hash value. There is no special mention when users use these hash functions. When SHAKE is used, SHAKE128 should be used for security level I and SHAKE256 for security levels III and V. On the other hand, we describe the case where an arbitrary-length hash value is generated using a hash function that outputs a fixed-length hash value, such as SHA256. Depending on the implementation environment, it may be impossible to use SHAKE, and in such cases, this is resolved by using MGF1. (See Subsection 2.3 to refer to MGF1.) In this case, $\mathsf{Hash}(\mathbf{M}||r)$ in the Sign and Verify algorithms would perform MGF1($\mathbf{M}||r$). When SHA-2 is used, SHA256, SHA384, and SHA512 should be used according to security level I, III, and V, respectively.

4.6 Note on the Generation of Random Finite Field Elements by using hash functions

Many random finite field elements are used when generating keys and sampling vinegar variables. For example, since an element of $\mathbb{F}_q[x]/(f)$ can be represented as a matrix $\mathbb{F}_q^{\ell\times\ell}$, the Expand functions in the KeyGen algorithm would generate elements of the finite field $\mathbb{F}_q$. A random bit sequence is generated using a hash function and retrieved for every $\lceil \log_2 q \rceil$ bit to generate these elements. By dividing a hash value into $\lceil \log_2 q \rceil$ bits, a sequence of random numbers in the range of $[0, 2^{\lceil \log_2 q \rceil})$ can be obtained. Even so, in the range of $[0, 2^{\lceil \log_2 q \rceil})$, q is the only number that does not belong to $\mathbb{F}_q$. Therefore, when q is obtained from the sequence of random numbers, q should be skipped and not chosen. Also, when obtaining the element of $\mathbb{F}_q^m$, the first m numbers that are non q values should be selected. This method, called rejection sampling, does not result in constant time, and we describe how to make it constant time in the following subsection.

4.6.1 Constant-time Rejection Sampling

Since q is an odd prime in QR-UOV, we use a rejection sampling for random $\mathbb{F}_q$ elements in our reference implementation. The rejection sampling may seem to be incompatible with the constant-time implementation, however, they can be practically compatible when the maximum number of elements to sample is known in advance. Algorithm 4 is a conceptual sketch of the constant-time rejection sampling with failure probability at most $2^{-\lambda}$. Although we do not use Algorithm 4 in our reference implementation, it should be adopted for applications where the side-channel attacks are critical. $\tau_{q,\lambda}(n)$ in Algorithm 4 is the threshold for how many or more uniform random numbers in $\{0, \ldots, 2^{\lceil \log_2 q \rceil} - 1\}$ are required for n uniform random sampling from $\mathbb{F}_q$. $\tau_{q,\lambda}(n)$ can be written as

$$\tau_{q,\lambda}(n) := \min\{t \in \mathbb{N} \mid P(n, t, q/2^{\lceil \log_2 q \rceil}) \leq 2^{-\lambda}\}.$$

$P(n, t, p)$ is the cumulative binomial distribution,

$$P(n, t, p) := \sum_{i=0}^{n-1} \binom{t}{i} p^i (1-p)^{t-i} = I_{1-p}(t-n+1, n),$$

which means the probability of less than n successes in t independent Bernoulli trials of success probability p. $I_z(a, b)$ is called the regularized incomplete beta function, which is suitable for numerical evaluation. The threshold $\tau_{q,\lambda}(n)$ may not be accurately evaluated on the fly, however there is no problem in applying an appropriate upper bound or pre-calculated value in practice.

4.7 Note on Basic Linear Algebra

Like other UOV schemes, QR-UOV also requires solving a system of linear equations to generate a signature. A tremendous amount of research exists

Algorithm 4 Constant-time Rejection Sampling for $\mathbb{F}_q^n$

Input: parameters (q, λ, n)
Output: a random sequence $(v_0, \ldots, v_{n-1}) \in \mathbb{F}_q^n$

$\tau \leftarrow \tau_{q,\lambda}(n)$,
allocate $(v_0, \ldots, v_{\tau-1})$, ▷ variables to store random elements
$j \leftarrow 0$,
$k \leftarrow \tau - 1$,
for $i = 1$ to τ **do**
  $r \xleftarrow{\$} \{0, \ldots, 2^{\lceil \log_2 q \rceil} - 1\}$,
  if $r \in \{0, \ldots, q-1\}$ **then** ▷ needs to be constant-time
    $v_j \leftarrow r,\ j \leftarrow j + 1$,
  else
    $v_k \leftarrow r,\ k \leftarrow k - 1$,
  end if
end for ▷ $\Pr[k < n-1] < 2^{-\lambda}$
return $(v_0, \ldots, v_{n-1})$.

on algorithms for solving linear equations, including well-known constant-time implementations[CKY21, BCH+23]. Although we did not use such an algorithm in our reference implementation, it should be employed for critical applications.

5 Implementation Details

In this section, we first describe how the algebra used in QR-UOV is represented and how its arithmetic operations are implemented in software. Then, we describe the way how public keys, private keys, and signatures for QR-UOV are stored in our implementation.

Of course a naive approach would be storing each finite field element of the matrix representation as it is, but a more refined method is provided in this section, that takes advantage of QR-UOV features.

5.1 Representation of Algebra

5.1.1 Representation of Finite Field Elements

In our implementation, each element in $\mathbb{F}_q$ is an integer in the range of $[0, q)$, represented as a bit string of $\lceil \log_2 q \rceil$ length. However, except in the middle of solving equation in the QR-UOV signature algorithm, each matrix belonging to $\mathcal{A}_f$ is represented in another way described in Subsubsection 5.1.2.

5.1.2 Representation of Polynomial Matrices of Quotient Ring

Suppose, the order of finite field q, the block size of the representation matrices ℓ, and an irreducible polynomial $f \in \mathbb{F}_q[x]$ with $\deg f = \ell$, are fixed.

Each $g \in \mathbb{F}_q[x]/(f)$ has ℓ coefficients. Thus it can be represented as the form of ℓ concatenated finite field elements, in length of $\lceil \log_2 q \rceil \cdot \ell$ bits. Since Φ_g^f is derived from g, it is also represented exactly in the same way.

5.2 Implementation of arithmetic operations

5.2.1 Arithmetic over Finite Fields

For addition, substraction, and multiplication over $\mathbb{F}_q$, our implementation simply computes them with the remainder operation. For inversion over $\mathbb{F}_q$, a look-up table is prepared and referenced, instead of complicated computation.

5.2.2 Arithmetic over Polynomial Matrices of Quotient Ring

For addition, substraction, and multiplication over $\mathbb{F}_q[x]/(f)$ or $\mathcal{A}_f$, our implementation simply computes them with the remainder operation.

Another option to implement these operations is to prepare and reference a look-up table for these operations. This way accelerates these operations, but requires a sufficiently large amount of memory.

5.2.3 Omission of Some Matrix Symmetrization

Actually, except in the middle of solving equation in the QR-UOV signature algorithm, both Φ_g^f and $W\Phi_g^f$ are represented as the same value that represents the corresponded g in our implementation. This is because all matrix operations involving $W\mathcal{A}_f$ used in QR-UOV are possible to be performed without actual symmetrization of $\mathcal{A}_f$, such as Proposition 1 in [FIKT21].

Only in the middle of solving equation, each $W\Phi_g^f$ is to be expanded in the symmetrized $\ell \times \ell$ matrix.

5.3 Representation of Keys and Signature

5.3.1 Public Key

The compressed form of a public key in QR-UOV is $\mathbf{pk}$, a pair of seed $\mathsf{seed}_{\mathsf{pk}}$ and $P_{i,3}$ $(i = 1, \ldots, m)$.

$\mathsf{seed}_{\mathsf{pk}}$ is stored as an octet string.

For $P_{i,3}$, each block matrix $W\Phi_g^f$ in $P_{i,3}$ is represented as the same way as g, in the range of $\lceil \log_2 q \rceil \cdot \ell$, as described in Subsubsection 5.1.2. Note that each $P_{i,3}$ $(i = 1, \ldots, m)$ is a symmetric matrix, so only the upper triangular part are needed to be stored. Therefore $P_{i,3}$ is stored as an octet string converted from a bit string of length $(\lceil \log_2 q \rceil \cdot \ell) \cdot (m(m+\ell)/(2\ell^2)) \cdot m$.

Remember that the full matrix representation of a public key P_i $(i = 1, \ldots, m)$ is composed of matrices $P_{i,1}, P_{i,2}, P_{i,3}$ $(i = 1, \ldots, m)$. In the middle of calculation, $P_{i,1}$ and $P_{i,2}$ are generated from the seed $\mathsf{seed}_{\mathsf{pk}}$, in the way described in Subsection 4.6. Each block matrix $W\Phi_g^f$ in $P_{i,1}$ and $P_{i,2}$ is represented in the same manner as that in $P_{i,3}$.

5.3.2 Private Key

The compressed form of a secret key in QR-UOV is only the secret seed $\mathsf{seed}_{\mathsf{sk}}$.

S' is generated from the seed $\mathsf{seed}_{\mathsf{sk}}$, in the way described in Subsection 4.6.

$F_{i,1}$ and $F_{i,2}$ $(i = 1, \ldots, m)$ are computed in the QR-UOV signature algorithm. They depends on $P_{i,1}, P_{i,2}$ $(i = 1, \ldots, m)$ generated from the public seed $\mathsf{seed}_{\mathsf{pk}}$, and S' generated from the secret seed $\mathsf{seed}_{\mathsf{sk}}$. Note that $P_{i,3}$ $(i = 1, \ldots, m)$ is not required.

Each block matrix $W\Phi_g^f$ in $S', F_{i,1}$ and $F_{i,2}$ $(i = 1, \ldots, m)$ is represented as the same way as the corresponded g, in the range of $\lceil \log_2 q \rceil \cdot \ell$, as described in Subsubsection 5.1.2.

5.3.3 Signature

A signature σ is a pair of salt $r \in \{0,1\}^\lambda$, and $\mathbf{s} \in \mathbb{F}_q^n$.

r is simply stored as an octet string converted from a bit string of length λ.

$\mathbf{s}$ is represented as a sequence of n finite field elements in $\mathbb{F}_q$. Thus, $\mathbf{s}$ can be stored as an octet string that is converted from a bit string of length $\lceil \log_2 q \rceil \cdot \ell$.

Actually, $\mathbf{s}$ is represented in the same manner as polynomial matrices of quotient ring. Assume $\frac{n}{\ell}$ matrices of size $\ell \times \ell$ belonging to $\mathcal{A}_f$ are given. By concatenating them vertically and ignoring the $2, \ldots, \ell$ th columns (i.e. leaving only the 1st column), it can be regarded as a vector belonging to $\mathbb{F}_q^n$.

6 Performance Analysis

6.1 Key and Signature Sizes

We here estimate the public key and signature size for the proposed parameter sets in Subsection 4.4. From the discussion in Section 4, the size of the public key, the secret key, and the signature size are given as follows

- public key: $\left(\lceil \log q \rceil \cdot \frac{m^2(m+\ell)}{2\ell} + 2\lambda \right)$ bits
- secret key: 2λ bits
- signature: $(\lceil \log q \rceil \cdot n + \lambda)$ bits

Note that the public and secret keys and signature include a seed and salt, respectively. As mentioned in Subsection 4.4, we set the security parameter λ as 128, 192, and 256 for the security level I, III, V, respectivly. Indeed, Table 2 computes the public key and signature size of the proposed parameter sets according to the above formulae.

Table 2: The public key and signature size of the parameter sets of QR-UOV proposed in Subsection 4.4

SL	(q, v, m, ℓ)	public key (B)	signature (B)
I	$(7, 740, 100, 10)$	20,657	331
	$(31, 165, 60, 3)$	23,657	157
	$(31, 600, 70, 10)$	12,282	435
	$(127, 156, 54, 3)$	24,271	200
III	$(7, 1100, 140, 10)$	55,173	489
	$(31, 246, 87, 3)$	71,007	232
	$(31, 890, 100, 10)$	34,423	643
	$(127, 228, 78, 3)$	71,915	292
V	$(7, 1490, 190, 10)$	135,439	662
	$(31, 324, 114, 3)$	158,453	306
	$(31, 1120, 120, 10)$	58,564	807
	$(127, 306, 105, 3)$	173,708	392

6.2 Performance on the NIST Reference Platform

Table 3 shows the timing data of experiments on the NIST reference platform for all parameter sets in Table 2. The target is a software implementation written in C, and it does not use special processor instructions. The environment is as follows.

Processor: AMD EPYC 7763.
Clock Speed: Boost Clock : Up to 3.5GHz, Base Clock: 2.45GHz.
Memory: 128GB (32GB RDIMM, 3200MT/s, Dual Rank, 8Gb base x4)
Operating System: Linux 5.19.0-41-generic, gcc version 11.3.0.
Measurement Software: supercop-20221122.

Table 3: Timing data on the NIST Reference Platform (Mcycles)

category	(q, v, m, ℓ)	keygen	sign	verify
I	$(127, 156, 54, 3)$	96.381	64.885	13.607
	$(31, 165, 60, 3)$	124.631	80.666	15.272
	$(31, 600, 70, 10)$	360.949	337.919	73.979
	$(7, 740, 100, 10)$	1129.934	997.764	168.411
III	$(127, 228, 78, 3)$	370.704	245.199	45.522
	$(31, 246, 87, 3)$	564.975	362.548	52.448
	$(31, 890, 100, 10)$	1589.343	1443.240	242.067
	$(7, 1100, 140, 10)$	5131.154	4493.868	535.040
V	$(127, 306, 105, 3)$	1172.189	755.475	104.209
	$(31, 324, 114, 3)$	1632.895	1021.670	112.420
	$(31, 1120, 120, 10)$	3628.424	3269.314	437.942
	$(7, 1490, 190, 10)$	15695.164	13226.016	1212.897

6.3 Performance on Other Platforms

Table 4 shows the results of an almost portable implementation for 64-bit environments. This implementation is also written in C. It does not use special processor instructions, but it ignores the 32-bit environment. The experimental environment is exactly the same as described above.

Table 4: Timing data on the NIST Reference Platform (Mcycles)

category	(q, v, m, ℓ)	keygen	sign	verify
I	$(127, 156, 54, 3)$	16.700	13.419	10.575
	$(31, 165, 60, 3)$	20.223	15.813	11.614
	$(31, 600, 70, 10)$	93.984	92.480	73.814
	$(7, 740, 100, 10)$	177.911	167.711	99.755
III	$(127, 228, 78, 3)$	65.263	52.290	37.159
	$(31, 246, 87, 3)$	85.616	65.286	42.450
	$(31, 890, 100, 10)$	387.796	362.721	245.240
	$(7, 1100, 140, 10)$	905.595	822.727	385.265
V	$(127, 306, 105, 3)$	217.373	158.856	81.309
	$(31, 324, 114, 3)$	233.036	168.576	87.673
	$(31, 1120, 120, 10)$	826.049	783.495	474.469
	$(7, 1490, 190, 10)$	2528.767	2220.364	844.445

Table 5 shows the results for an implementation that uses avx2. It seems that a little more optimization can be done.

Table 5: Timing data on the avx2 Platform (Mcycles)

category	(q, v, m, ℓ)	keygen	sign	verify
I	$(127, 156, 54, 3)$	30.885	24.823	13.539
	$(31, 165, 60, 3)$	31.691	25.217	15.973
	$(31, 600, 70, 10)$	117.225	135.849	68.947
	$(7, 740, 100, 10)$	355.028	361.728	144.955
III	$(127, 228, 78, 3)$	125.521	98.376	47.636
	$(31, 246, 87, 3)$	206.605	153.006	53.490
	$(31, 890, 100, 10)$	553.788	573.433	232.156
	$(7, 1100, 140, 10)$	1552.650	1555.131	524.886
V	$(127, 306, 105, 3)$	293.487	221.341	108.650
	$(31, 324, 114, 3)$	468.632	337.483	119.098
	$(31, 1120, 120, 10)$	1004.973	1074.835	433.574
	$(7, 1490, 190, 10)$	4484.707	4254.736	1169.402

Table 6 shows the results for an implementation that uses avx512. Only avx512 was measured in the following environment, due to the experimental reason.

Processor: Intel Xeon W 2223.
Clock Speed: Boost Clock : Up to 3.90GHz, Base Clock: 3.60GHz.
Memory: 32GB (32GB DDR4, 2600MT/s, Dual Channel, 16Gb base x2)
Operating System: Linux 5.15.90.1-microsoft-standard-WSL2, gcc version 11.3.0.
Measurement Software: supercop-20221122.

Table 6: Timing data on the avx512 Platform (Mcycles)

category	(q, v, m, ℓ)	keygen	sign	verify
I	$(127, 156, 54, 3)$	92.239	78.676	35.482
	$(31, 165, 60, 3)$	130.975	98.611	38.300
	$(31, 600, 70, 10)$	271.039	304.873	150.258
	$(7, 740, 100, 10)$	793.458	800.165	310.911
III	$(127, 228, 78, 3)$	279.735	220.191	100.650
	$(31, 246, 87, 3)$	551.029	378.887	112.647
	$(31, 890, 100, 10)$	1164.509	1159.963	442.188
	$(7, 1100, 140, 10)$	3956.843	3632.370	980.261
V	$(127, 306, 105, 3)$	982.046	685.904	220.980
	$(31, 324, 114, 3)$	1083.549	735.542	246.744
	$(31, 1120, 120, 10)$	2030.460	2068.606	792.030
	$(7, 1490, 190, 10)$	10558.854	8596.815	1964.954

7 Expected Security Strength

In this section, we first provide underlying problems and the statement for our security proof. We second estimate the complexity of considerable attacks against our proposed parameter sets in Subsection 4.4. See Section 8 for the details of each attack.

7.1 Underlying Problems and Security Proof

This subsection discusses the security of our QR-UOV described in Section 4. After introducing two assumptions based on which the security proof of QR-UOV can be constructed and the standard security definition, we show the statement of the EUF-CMA security of QR-UOV.

We first introduce two problems for the security proof of QR-UOV as follows:

Definition 1 (UOV problem)**.** *We let* $\mathsf{MQ}_{q,n,m}$ *the set of random quadratic maps with* $n = v+o$ *variables and* m *equations over* $\mathbb{F}_q$ *and let* $\mathsf{UOV}_{q,v,o,m}$ *the set of public key maps of the plain UOV with* v *vinegar variables,* o *oil variables, and* m *equations over* $\mathbb{F}_q$*. The UOV problem asks to distinguish a random quadratic system from a UOV public key. If we let* $\mathcal{A}$ *be a UOV distinguisher algorithm, then we say the distinguishing advantage of* $\mathcal{A}$ *is*

$$\begin{aligned}&\mathsf{Adv}^{\mathsf{UOV}}_{q,v,o,m}(\mathcal{A}) = \\ &|\Pr[\mathcal{A}(\mathcal{P}) = 1 \mid \mathcal{P} \leftarrow \mathsf{MQ}_{q,(v+o),m}] - \Pr[\mathcal{A}(\mathcal{P}) = 1 \mid \mathcal{P} \leftarrow \mathsf{UOV}_{q,v,o,m}]|.\end{aligned}$$

Definition 2 (QR-MQ problem)**.** *Let* f *be an irreducible polynomial with* $\deg f = \ell$ *and* $N = n/\ell$*. We then denote by* $\mathsf{QR}_{q,n,m,\ell}$ *the set of quadratic maps constructed as follows*

$$\mathsf{QR}_{q,n,m,\ell} = \left\{ \left(\mathbf{x}^\top P_1 \mathbf{x}, \ldots, \mathbf{x}^\top P_m \mathbf{x}\right) : \mathbb{F}_q^n \to \mathbb{F}_q^m \mid P_1, \ldots, P_m \in W^{(N)} \mathcal{A}_f^{N,N} \right\}.$$

For a randomly chosen $\mathcal{P} \in \mathsf{QR}_{q,n,m,\ell}$ *and* $\mathbf{t} \in \mathbb{F}_q^m$*, the QR-MQ problem asks to compute* $\mathbf{s}$ *such that* $\mathcal{P}(\mathbf{s}) = \mathbf{t}$*. If we let* $\mathcal{A}$ *be an adversary, then we say that the advantage of* $\mathcal{A}$ *against the QR-MQ problem is*

$$\mathsf{Adv}^{\mathsf{QR\text{-}MQ}}_{q,n,m,\ell}(\mathcal{A}) = \Pr[\mathcal{P}(\mathbf{s}) = \mathbf{t} \mid \mathcal{P} \leftarrow \mathsf{QR}_{q,n,m,\ell}, \mathbf{t} \leftarrow \mathbb{F}_q^m, \mathbf{s} \leftarrow \mathcal{A}(\mathcal{P}, \mathbf{t})].$$

We prove the security of QR-UOV below assuming the advantages against the above two problems are negligible. The first assumption is originally utilized for the security of the plain UOV and thus seems relatively well understood. By contrast, the second assumption is inherent in QR-UOV. Therefore, it is one of the important tasks to correctly evaluate the difficulty of the QR-MQ problem.

Subsequently, we give the definition of the EUF-CMA security, which is the standard security definition for digital signature schemes.

Definition 3 (EUF-CMA security)**.** *Let* $\mathcal{O}$ *be a random oracle, and let* $\mathcal{A}$ *be an adversary. We say the advantage of* $\mathcal{A}$ *against the EUF-CMA game of a*

signature scheme $\mathrm{DSS} = (\mathrm{KeyGen}, \mathrm{Sign}^{\mathcal{O}}, \mathrm{Verify}^{\mathcal{O}})$ *in the random oracle model is*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathrm{DSS}}(\mathcal{A}) =$$
$$\Pr[\mathrm{Verify}^{\mathcal{O}}(\mathsf{pk}, m, \sigma) = 1 \mid (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathrm{KeyGen}(), (m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}, \mathrm{Sign}^{\mathcal{O}}(\mathsf{sk}, \cdot)}(\mathsf{pk})],$$

where $\mathsf{Sign}^{\mathcal{O}}(\mathsf{sk}, \cdot)$ *was not queried on input* m*. We say* DSS *is EUF-CMA secure if its advantage is negligible for any efficient adversary in the security parameter.*

We then show the EUF-CMA security of the QR-UOV signature scheme. This theorem is proven mainly based on Proposition 5.3 in [KX22] by Kosuge and Xagawa.

Theorem 2 (QR-MQ and UOV $\Rightarrow$ EUF-CMA)**.** *For any quantum EUF-CMA adversary* $\mathcal{A}_{\mathsf{cma}}$ *of QR-UOV issuing at most* q_{sign} *classical queries to the signing oracle and* q_{qro} *(quantum) random oracle queries to* Hash*, there exist adversaries* $\mathcal{B}$ *and* $\mathcal{B}'$ *against the* $UOV_{q^\ell, v/\ell, o/\ell, m}$ *and* $QR\text{-}MQ_{q,(v+m),m,\ell}$ *assumptions respectively*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{QR\text{-}UOV}}(\mathcal{A}_{\mathsf{cma}}) \leq (2q_{\mathsf{qro}} + 1)^2 \left(\mathsf{Adv}^{\mathsf{UOV}}_{q^\ell, v/\ell, o/\ell, m}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{QR\text{-}MQ}}_{q,(v+m),m,\ell}(\mathcal{B}')\right) + \frac{3}{2} q'_{\mathsf{sign}} \sqrt{\frac{q'_{\mathsf{sign}} + q_{\mathsf{qro}} + 1}{2^\lambda}} + 2\left(q_{\mathsf{sign}} + q_{\mathsf{qro}} + 2\right) \sqrt{\frac{q'_{\mathsf{sign}} - q_{\mathsf{qro}}}{2^\lambda}},$$

where q'_{sign} *is a bound on the total number of queries to* Hash *in all the signing queries and the running time of* $\mathcal{B}$ *and* $\mathcal{B}'$ *is about that of* $\mathcal{A}_{\mathsf{cma}}$*.*

In the following, we give the proof of Theorem 2. Our proof of Theorem 2 is mainly depending on a result by Kosuge and Xagawa [KX22], which shows the EUF-CMA security of QR-UOV assuming the difficulty of the INV (non-INVertibility) game of QR-UOV. If we denote by $\mathsf{QR}_{q,v,m,\ell}$ the set of public key maps of QR-UOV with parameters (q, v, m, ℓ), then the advantage of $\mathcal{A}$ against the INV game of QR-UOV is given by

$$\mathsf{Adv}^{\mathsf{INV}}_{\mathrm{DSS}}(\mathcal{A}) = \Pr[\mathcal{P}(\mathbf{x}) = \mathbf{t} \mid \mathcal{P} \leftarrow \mathsf{QR}_{q,v,m,\ell}, \mathbf{t} \leftarrow \mathbb{F}_q^m, \mathbf{x} \leftarrow \mathcal{A}(\mathcal{P}, \mathbf{t})].$$

Proposition 5.3 in [KX22] originally shows the EUF-CMA security of the plain UOV signature scheme with a modification proposed by Sakumoto et al. [SSH11], and they claim that they can apply this proposition to QR-UOV with the modification by Sakumoto et al. Note that we here describe the proposition as the one for QR-UOV and change some notations for consistency.

Lemma 2 (Proposition 5.3 in [KX22], INV $\Rightarrow$ EUF-CMA (Modified UOV Signature))**.** *For any quantum EUF-CMA adversary* $\mathcal{A}_{\mathsf{cma}}$ *of QR-UOV issuing at most* q_{sign} *classical queries to the signing oracle and* q_{qro} *(quantum) random*

oracle queries to Hash*, there exist an INV adversary* $\mathcal{B}_{\mathsf{inv}}$ *of QR-UOV such that*

$$\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathsf{QR\text{-}UOV}}(\mathcal{A}_{\mathsf{cma}}) \leq (2q_{\mathsf{qro}}+1)^2\,\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{uov}}}(\mathcal{B}_{\mathsf{inv}}) + \frac{3}{2}q'_{\mathsf{sign}}\sqrt{\frac{q'_{\mathsf{sign}}+q_{\mathsf{qro}}+1}{2^\lambda}} + 2\,(q_{\mathsf{sign}}+q_{\mathsf{qro}}+2)\sqrt{\frac{q'_{\mathsf{sign}}-q_{\mathsf{qro}}}{2^\lambda}},$$

where q'_{sign} *is a bound on the total number of queries to* Hash *in all the signing queries and the running time of* $\mathcal{B}_{\mathsf{inv}}$ *is about that of* $\mathcal{A}_{\mathsf{cma}}$*.*

Subsequently, we discuss the non-invertibility of QR-UOV. Before providing the proof, we prepare a lemma which shows a bijection from the key space of QR-UOV to that of the plain UOV over the extension field. We call this transformation a *pull-back method*.

Lemma 3. *For any parameters* (q, v, m, ℓ) *of QR-UOV and an irreducible polynomial* f *with* $\deg f = \ell$*, there exists a one-to-one mapping from the key space of QR-UOV with parameter* (q, v, m, ℓ) *into that of the plain UOV with* v/ℓ *vinegar variables,* m/ℓ *oil variables, and* m *equations over* $\mathbb{F}_{q^\ell}$*.*

Proof. We here show the correctness of the statement by constructing such a one-to-one map representing the keys of QR-UOV as those of the plain UOV over $\mathbb{F}_q[x]/(f) \cong \mathbb{F}_{q^\ell}$. For each representation matrix P_k of the public key of QR-UOV, we can take ℓ matrices $\bar{P}_k^{(0)}, \ldots, \bar{P}_k^{(\ell-1)} \in \mathbb{F}_q^{N\times N}$ satisfying

$$P_k = \sum_{i=0}^{\ell-1}\left(\bar{P}_k^{(i)} \otimes W\Phi^f_{x^i}\right),$$

due to the structure of the QR-UOV public key. We then can define an $N \times N$ matrix $\bar{P}_k$ over $\mathbb{F}_q[x]/(f)$ as follows:

$$\bar{P}_k = \sum_{i=0}^{\ell-1} x^i \bar{P}_k^{(i)}.$$

By using the same way, we can construct $\bar{F}_1, \ldots, \bar{F}_m$ and $\bar{S}$ corresponding to the secret key $F_1, \ldots, F_m$ and S as follows:

$$F_k = \sum_{i=0}^{\ell-1}\left(\bar{F}_k^{(i)} \otimes W\Phi^f_{x^i}\right) \;\Rightarrow\; \bar{F}_k = \sum_{i=0}^{\ell-1} x^i \bar{F}_k^{(i)},$$
$$S = \sum_{i=0}^{\ell-1}\left(\bar{S}^{(i)} \otimes \Phi^f_{x^i}\right) \;\Rightarrow\; \bar{S} = \sum_{i=0}^{\ell-1} x^i \bar{S}^{(i)}.$$

Then, it holds $\bar{P}_k = \bar{S}^\top \bar{F}_k \bar{S}$ from $P_k = S^\top F_k S$, and $\bar{F}_k$ has the form as in (3). Thus, these set of $\bar{P}_k$, $\bar{F}_k$, and $\bar{S}$ can be seen as the keys of the plain UOV with

N variables and m equations over the extension field $\mathbb{F}_q[x]/(f)$. This transformation is clearly a bijective map from the key space $\left(\{P_k\}_{k\in[m]}, \{F_k\}_{k\in[m]}, S\right)$ of QR-UOV into the key space $\left(\{\bar{P}_k\}_{k\in[m]}, \{\bar{F}_k\}_{k\in[m]}, \bar{S}\right)$ of the plain UOV over the extension field $\mathbb{F}_q[x]/(f)$. □

We then show the security of the INV game of QR-UOV assuming the difficulty of the UOV and QR-MQ problems.

Lemma 4 (UOV and QR-MQ ⇒ INV)**.** *For any quantum INV adversary $\mathcal{A}_{\mathsf{inv}}$ of $\mathsf{HaS}[\mathsf{T}_{\mathsf{uov}}, \mathsf{H}]$, there exist adversaries $\mathcal{B}$ and $\mathcal{B}'$ against the $UOV_{q^\ell,v/\ell,o/\ell,m}$ and $QR\text{-}MQ_{q,(v+m),m,\ell}$ assumptions respectively*

$$\mathsf{Adv}^{\mathsf{INV}}_{\mathsf{T}_{\mathsf{uov}}}(\mathcal{A}_{\mathsf{inv}}) \leq \mathsf{Adv}^{\mathsf{UOV}}_{q^\ell,v/\ell,o/\ell,m}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{QR\text{-}MQ}}_{q,(v+m),m,\ell}(\mathcal{B}'),$$

where the running time of $\mathcal{B}$ and $\mathcal{B}'$ is about that of $\mathcal{A}_{\mathsf{inv}}$.

Proof. Let Game_0 be $\mathcal{A}_{\mathsf{inv}}$'s INV game against QR-UOV. We then have the equation $\Pr[\mathsf{Game}_0() = 1] = \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{q,v,m,\ell}(\mathcal{A}_{\mathsf{inv}})$. Let Game_1 be just the QR-MQ game, which is the same as Game_0 except that the adversary receives a randomly chosen map in $\mathsf{QR}_{q,n,m,\ell}$ instead of the public key of QR-UOV. This Game_0 and Game_1 can be seen as an adversary against a problem asking to distinguish the public key of QR-UOV and a randomly chosen map in $\mathsf{QR}_{q,n,m,\ell}$, and this problem is equivalent to the UOV problem with v/ℓ vinegar variables, o/ℓ oil variables, and m equations over the extension field $\mathbb{F}_{q^\ell}$ from Lemma 3. We then can clearly generate adversaries $\mathcal{B}$ and $\mathcal{B}'$ against the $UOV_{q^\ell,v/\ell,o/\ell,m}$ and $QR\text{-}MQ_{q,(v+m),m,\ell}$ assumptions, that run in about the time of $\mathcal{A}_{\mathsf{inv}}$, and we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{UOV}}_{q^\ell,v/\ell,o/\ell,m}(\mathcal{B}) &= |\Pr[\mathsf{Game}_0() = 1] - \Pr[\mathsf{Game}_1() = 1]| \\ &= |\mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{q,v,m,\ell}(\mathcal{A}_{\mathsf{inv}}) - \mathsf{Adv}^{\mathsf{QRMQ}}_{q,(v+m),m,\ell}(\mathcal{B}')|.\end{aligned}$$

□

From Lemma 2 and 4, we clearly obtain the statement of Theorem 2.

Remark 1. *In [CDP23], Chatterjee et al. claimed that there exist some issues in the EUF-CMA security proof given by Sakumoto et al. [SSH11]. Our QR-UOV uses the modification of the signature generations for the security proof used in the proof by Sakumoto et al. However, our security reduction is dependent on a different result by Kosuge and Xagawa [KX22]. Thus, the result by Chatterjee et al. does not affect our security proof.*

Remark 2. *In [Has22], Hashimoto provides a new way of constructing QR-UOV by a smaller UOV over an extension field. We note that the way of transforming the public and secret keys of QR-UOV into those of the plain UOV used in [Has22] is equivalent to the one described in Lemma 3.*

Remark 3. *In the proposed scheme, we set the length of the salt included in signatures as 128, 192, and 256 bits for the security levels I, III, and V, respectively. We can take a more conservative choice of the salt length to make the additive terms* $\frac{3}{2}q'_{\mathsf{sign}}\sqrt{\frac{q'_{\mathsf{sign}}+q_{\mathsf{qro}}+1}{2^{\lambda}}}$ *and* $2\left(q_{\mathsf{sign}}+q_{\mathsf{qro}}+2\right)\sqrt{\frac{q'_{\mathsf{sign}}-q_{\mathsf{qro}}}{2^{\lambda}}}$ *in Theorem 2 negligible. We first consider the size of* q'_{sign} *which denotes a bound on the total number of queries to* Hash *in all the signing queries in Theorem 2. In the proof of Theorem 2 in [KX22], we have to set this* q'_{sign} *such that the failure probability of generating signatures* q_{sign} *times with a bound* q'_{sign} *on the number of queries to* Hash*. From the expected number of queries to* Hash *for one time signature generation, we can estimate* q'_{sign} *as* $\approx 2^{100}$ *to make the failure probability* $\approx 2^{-128}$*. Then, the length of salt is determined as approximately* 500 *bits to make two additive terms negligible. This can be seen as a tradeoff between the length of signature and the security. Note that there has not been proposed any attack that can take advantage of the loss of the security reduction.*

7.2 Security Estimation of the Proposed Parameters

In this subsection, we confirm that the proposed parameters in Subsection 4.4 satisfy security levels I, III, and V of the NIST PQC project by estimating the complexity of considerable attacks on QR-UOV described in Section 8 as seen in Table 7 and 8.

As stated in Subsection 7.1, the EUF-CMA security of QR-UOV can be reduced to the difficulty of the QR-MQ problem and the UOV problems with parameters $(q^{\ell}, v/\ell, m/\ell, m)$, namely UOV with v/ℓ vinegar variables, m/ℓ oil variables, and m equations over $\mathbb{F}_{q^{\ell}}$. We here consider the hybrid approach [BFP09] with Wiedemann XL (WXL) [YCBC07] and the polynomial XL (PXL) [FK21] as attackers against the (QR-)MQ problem and the Kipnis-Shamir [KS98], reconciliation [DYC+08], intersection [Beu21], and rectangular MinRank [Beu21] attacks as attackers against the UOV problem with parameters $(q^{\ell}, v/\ell, m/\ell, m)$. (See Subsection 8.3 for the relationship between the difficulty of the QR-MQ problem and the plain MQ problem.) In addition to these attacks, we consider the claw finding attack and the complexity of these attacks is evaluated in Table 7.

Furthermore, we can clearly perform the key recovery attacks on QR-UOV by simply regarding QR-UOV as the plain UOV with parameters (q, v, m, m). In Table 8, we estimate the complexity of three key recovery attacks, the Kipnis-Shamir, reconciliation, and intersection attacks, on the plain UOV with parameters (q, v, m, m) for each proposed parameter set. (See Subsection 8.5 for the reason that the rectangular MinRank is not applicable to the plain UOV with parameters (q, v, m, m).) Then, one can confirm that for each attack, the complexity on the UOV problem with parameters (q, v, m, m) is larger than or equal to that on the UOV problem with parameters $(q^{\ell}, v/\ell, m/\ell, m)$. Therefore, as mentioned above, our proposed parameters satisfy each claimed security level.

Here, security levels I, III, and V indicate that a classical attacker needs more than 2^{143}, 2^{207}, and 2^{272} classical gates to break the parameters, whereas

a quantum attacker needs more than 2^{61}, 2^{125}, and 2^{189} quantum gates, respectively, from the call for additional digital signature schemes [NIS22]. The number of gates required for each attack can be computed using

$$\#\text{gates} = \#\text{field multiplication} \cdot \left(2 \cdot (\log_2 q)^2 + \log_2 q\right).$$

In Table 7 and 8, for each parameter set, the upper entry shows the number of classical gates, whereas the lower entry shows the number of quantum gates. Furthermore, the values in bold indicate the complexity of the best attack against each parameter set. As a result, these tables show that the proposed parameters satisfy the requirements for each security level. One can confirm that our proposed parameters for security levels I and III also satisfy security levels II and IV, respectively.

Table 7: The complexity of the classical and quantum considerable attacks, the claw finding attack, and the Hashimoto's method with WXL (WXL) and PXL on the MQ problem, and the Kipnis-Shamir (KS), reconciliation (Recon.), intersection (Inter.), and rectangular MinRank (RM) attacks on UOV$(q^\ell, v/\ell, m/\ell, m)$ against the proposed parameter sets

SL	(q, v, m, ℓ)	Claw	WXL	PXL	KS	Recon.	Inter.	RM
I	$(7, 740, 100, 10)$	154	184	201	1793	**148**	1641	162
		154	**105**	136	908	148	869	162
	$(31, 165, 60, 3)$	162	163	152	531	**151**	343	153
		162	**117**	127	279	151	224	153
	$(31, 600, 70, 10)$	187	164	162	2600	**152**	2415	157
		187	**111**	134	1312	152	1251	157
	$(127, 156, 54, 3)$	202	160	**150**	718	164	460	158
		202	132	**128**	372	164	282	158
III	$(7, 1100, 140, 10)$	**211**	262	283	2693	219	2452	229
		211	**152**	188	1359	219	1287	229
	$(31, 246, 87, 3)$	229	226	**215**	801	221	508	220
		229	**171**	180	415	220	323	220
	$(31, 890, 100, 10)$	262	235	232	3890	**216**	3585	220
		262	**164**	193	1958	216	1851	220
	$(127, 228, 78, 3)$	287	222	**211**	1056	231	653	219
		287	182	**180**	542	227	390	219
V	$(7, 1490, 190, 10)$	281	354	384	3649	**277**	3291	292
		281	**213**	253	1838	277	1719	292
	$(31, 324, 114, 3)$	297	286	**279**	1055	283	658	**279**
		297	**218**	232	543	280	413	279
	$(31, 1120, 120, 10)$	311	280	**275**	4931	283	4540	290
		311	**197**	230	2479	283	2335	290
	$(127, 306, 105, 3)$	381	288	279	1414	291	851	**277**
		381	**237**	238	722	289	505	277

Table 8: The complexity of the classical and quantum considerable key recovery attacks on UOV(q, v, m, m), the Kipnis-Shamir (KS), reconciliation (Recon.), and intersection (Inter.) attacks, against the proposed parameter sets

SL	(q, v, m, ℓ)	KS	Recon.	Inter.
I	$(7, 740, 100, 10)$	1825	1350	2089
		928	891	1140
	$(31, 165, 60, 3)$	545	408	666
		287	314	450
	$(31, 600, 70, 10)$	2651	1368	2788
		1340	1038	1528
	$(127, 156, 54, 3)$	736	421	772
		383	348	527
III	$(7, 1100, 140, 10)$	2725	1980	3088
		1379	1302	1673
	$(31, 246, 87, 3)$	814	591	980
		423	451	653
	$(31, 890, 100, 10)$	3941	2001	4129
		1987	1516	2250
	$(127, 228, 78, 3)$	1073	553	596
		553	491	753
V	$(7, 1490, 190, 10)$	3681	2661	4167
		1858	1745	2253
	$(31, 324, 114, 3)$	1069	763	1278
		551	581	848
	$(31, 1120, 120, 10)$	4983	2502	5201
		2508	1893	2820
	$(127, 306, 105, 3)$	1431	782	1477
		732	644	997

8 Analysis of Attacks against QR-UOV

This section describes considerable attacks on QR-UOV. The rest of this section is organized as follows. Subsection 8.1 discusses the effect of irreducibility of the polynomial f constructing the quotient ring of QR-UOV. Subsection 8.2 explains the direct attack which directly finds a signature for a given message. Subsection 8.3 provides another way of reducing the public and secret keys of QR-UOV into those of the plain UOV over extension fields. Subsection 8.4 recall known key recovery attacks on the plain UOV. Subsection 8.5 shows that the rectangular MinRank attack [Beu21] is applicable to UOV over extension fields. Subsection 8.6 shows another way of attacking QR-UOV over the extension field as in the pull-back method given in Lemma 3.

8.1 Irreducibility of f

The public key of our QR-UOV is given as block matrices whose each component is an element of $W\mathcal{A}_f$. For the security of QR-UOV, we here discuss the relation between the irreducibility of polynomial f of $\mathcal{A}_f$ and the existence of transformation on symmetric matrices $W\Phi_g^f$ into a specific form matrix. Indeed, the security of BAC-UOV [SP20] whose public key is represented as block anti-circulant matrices was weakened by transforming anti-circulant matrices into a specific form with zero submatrices [FKI+20]. Therefore, we have to find f such that there exists no such a transformation on $W\Phi_g^f$.

In [FIKT21], they provide the following three theorems for the transformation on $W\Phi_g^f$ which show the suitability of an irreducible f for QR-UOV.

Theorem 3 (Theorem 4 in [FIKT21]). *Let $f \in \mathbb{F}_q[x]$ be a **reducible** polynomial with $\deg f = \ell$ and W be an invertible matrix such that every element of $W\mathcal{A}_f$ is a symmetric matrix. If $f = f_1 \cdots f_k$ ($k \in \mathbb{N}$), where $f_1, \ldots, f_k$ are distinct and irreducible, and $\deg f_1 \leq \cdots \leq \deg f_k$, then there exists an invertible matrix $L \in \mathbb{F}_q^{\ell\times\ell}$ and $i \in [\ell - 1]$ such that for any $X \in W\mathcal{A}_f$,*

$$L^\top X L = \begin{pmatrix} *_{i\times i} & 0_{i\times(\ell-i)} \\ 0_{(\ell-i)\times i} & *_{(\ell-i)\times(\ell-i)} \end{pmatrix}.$$

Theorem 4 (Theorem 5 in [FIKT21]). *With the same notation as in Theorem 3, if there exists $f' \in \mathbb{F}_q[x]$ such that $f'^2 \mid f$, there exists an invertible matrix $L \in \mathbb{F}_q^{\ell\times\ell}$ such that, for any $X \in W\mathcal{A}_f$,*

$$(L^\top X L)_{\ell\ell} = 0.$$

Theorem 5 (Theorem 2 in [FIKT21]). *Let $f \in \mathbb{F}_q[x]$ be an **irreducible** polynomial with $\deg f = \ell$ and W be an invertible matrix such that every element of $W\mathcal{A}_f$ is a symmetric matrix. Then, there is no invertible matrix $L \in \mathbb{F}_q^{\ell\times\ell}$ and $i, j \in [\ell]$ such that for any $X \in W\mathcal{A}_f$,*

$$(L^\top X L)_{ij} = 0.$$

Theorems 3 and 4 show that if f is reducible, for any $X \in W\mathcal{A}_f$, X can be transformed into a matrix with zero submatrices by multiplying an invertible matrix and its transposition from both sides. By contrast, Theorem 5 shows that if f is irreducible there exists no such a transformation on $W\Phi_g^f$. Therefore, we choose an irreducible polynomial as the f of $\mathcal{A}_f$ used in our proposed QR-UOV.

8.2 Claw Finding Attack

This subsection considers the claw finding attack on $\mathcal{P}(\mathbf{s}) = \mathsf{Hash}(\mathbf{M}||r)$, which is also called the birthday attack. We estimate the complexity of this attack according to the estimation in [BCC$^+$22, BCH$^+$23].

For a message $\mathbf{M}$, an attacker computes $\mathcal{P}(\mathbf{s}_i)$ for X inputs $\{\mathbf{s}_i\}_{i\in[X]}$ and compute $\mathsf{Hash}(\mathbf{M}||r_j)$ for Y salts $\{r_j\}_{j\in[Y]}$. If $XY = q^m$, then there is a collision with probability $\approx 1 - e^{-1}$, and the attacker can output the signature $(r_j, \mathbf{s}_i)$ for the message $\mathbf{M}$. In [BCC$^+$22, BCH$^+$23], they estimate the complexity of this attack considering the cost of multiplication and addition in $\mathbb{F}_q$ as follows:

$$2\left(q^m \cdot m \cdot 2^{17} \cdot \left(2 \cdot (\log_2 q)^2 + \log_2 q\right)\right)^{\frac{1}{2}}. \tag{8}$$

In this estimation, we suppose that computing Hash has a bit cost of 2^{17} and applying a fast enumeration algorithm [FT23] in $\mathbb{F}_q$ to evaluate $\mathcal{P}$ successively.

For the quantum claw finding attack, it is shown that attackers with limited time will prefer the classical attacks in [JS19]. Thus, in Subsection 7.2, we also esimate the complexity of the quantum claw finding attack by equation (8).

Remark 4 (Hash Collision Attack)**.** *A hash collision attack finds two inputs M and M' for the hush function* Hash *satisfying* $\mathsf{Hash}(\mathbf{M}) = \mathsf{Hash}(\mathbf{M}')$*. This attack does not work on the proposed scheme, since in the signature generation, the form of inputs for* Hash *is $\mathbf{M}||r$ with a message $\mathbf{M}$ and a salt r and this r is randomly chosen by the signer.*

8.3 Direct Attack

This subsection describes the direct attack, whose construction can be described as follows: Given public key $\mathcal{P}$ and a target $\mathbf{m} \in \mathbb{F}_q^m$, the direct attack tries to solve the MQ system $\mathcal{P}(\mathbf{s}) = \mathbf{m}$ to find a signature $\mathbf{s}$.

Given a quadratic polynomial system $\mathcal{P} = (p_1, \ldots, p_m)$ in n variables over $\mathbb{F}_q$ and $\mathbf{m} \in \mathbb{F}_q^m$, the direct attack algebraically solves the system $\mathcal{P}(\mathbf{x}) = \mathbf{m}$. We first explain the complexity of solving the MQ system with $n \leq m$ (overdetermined), second show a way of reducing the MQ system with $n > m$ (underdetermined) into a smaller overdetermined system, and third discuss the difficulty of the QR-MQ problem to which the security of QR-UOV is reduced in Theorem 2.

Overdetermined case We here provide a way of estimating the complexity of solving the overdetermined MQ system, since an underdetermined system can

be transformed into an overdetermined system by specifying $n - m$ variables without disturbing the existence of a solution with high probability. One of the best-known approaches for algebraically solving the quadratic system is the hybrid approach [BFP09], which randomly guesses k ($k = 0, \ldots, n$) variables before applying an MQ solver such as F4 [Fau99], F5 [Fau02], and XL [CKPS00]. The guessing process is repeated until a solution is obtained. The complexity of this approach with the Wiedemann XL (WXL) [YCBC07], which is a variant of XL, for a classical adversary is given by

$$\min_k \left(O\left(q^k \cdot 3 \cdot \binom{n-k+2}{2} \cdot \binom{d_{reg}+n-k}{d_{reg}}^2 \right) \right), \tag{9}$$

where d_{reg} is the so-called degree of regularity of the system. The degree of regularity d_{reg} for a certain class of polynomial systems called *semi-regular systems* [Bar04, BFS03, BFSY05] is known to be the degree of the first non-positive term in the following series [BFSY05, YC05]:

$$\frac{\left(1-z^2\right)^m}{\left(1-z\right)^{n-k+1}}. \tag{10}$$

Empirically, the public key system of UOV is considered to be a semi-regular system. Therefore, this series (10) can be used to estimate the degree of regularity. By using Grover's algorithm [Gro96], the complexity of a quantum direct attack is estimated as

$$\min_k \left(O\left(q^{k/2} \cdot 3 \cdot \binom{n-k+2}{2} \cdot \binom{d_{reg}+n-k}{d_{reg}}^2 \right) \right). \tag{11}$$

Furthermore, a new variant of the hybrid approach with XL, which is called polynomial XL (PXL), was proposed at 2021 [FK21]. This PXL reduces the complexity by performing Gaussian elimination on the matrix over a polynomial ring and the complexity of PXL for classical and quantum attackers is given by

$$O\left(k^2 \cdot \alpha \cdot \tbinom{n-k+d_{reg}}{d_{reg}} \cdot \tbinom{n+d_{reg}}{d_{reg}} + q^k \cdot \left(\alpha^2 \cdot \tbinom{k+d_{reg}}{d_{reg}} + \alpha^\omega \right) \right),$$
$$O\left(k^2 \cdot \alpha \cdot \tbinom{n-k+d_{reg}}{d_{reg}} \cdot \tbinom{n+d_{reg}}{d_{reg}} + q^{\frac{k}{2}} \cdot \left(\alpha^2 \cdot \tbinom{k+d_{reg}}{d_{reg}} + \alpha^\omega \right) \right),$$

respectively, where k is the number of guessed variables and $\omega = 2.37$ is the constant in the complexity of matrix multiplication. Furthermore, this α is given as

$$\sum_{d=0}^{d_{reg}} \max\left\{ \text{coeff}\left(\left(1-z\right)^{m-(n-k)} \left(1+z\right)^m, z^d \right), 0 \right\},$$

where $\text{coeff}(f, t)$ denotes the coefficient of t in f.

Table 9: Theoretical and experimental degree of regularity of public key system of QR-UOV obtained using the Magma algebra system [BCP97].

(q, v, m, ℓ, k)	theoretical d_{reg}	experimental d_{reg}
$(7, 24, 12, 3, 0)$	13	13
$(7, 24, 12, 3, 1)$	7	7
$(7, 24, 12, 3, 2)$	6	6
$(7, 30, 15, 3, 0)$	16	16
$(7, 30, 15, 3, 1)$	8	9
$(7, 30, 15, 3, 2)$	7	7

Underdetermined case We here explain a way of solving the underdetermined MQ system efficiently. Thomae and Wolf [TW12] proposed a technique for reducing the number of variables and equations when $n > m$. For $\alpha = \lfloor \frac{n}{m} \rfloor - 1$, they reduce the $(n-m+\alpha)$ variables and α equations and thereby obtain a quadratic system with $m-\alpha$ variables and equations. In [FNT21], Furue et al. improved Thomae and Wolf's technique supposing to guess values of k variables as in the hybrid approach, and Hashimoto proposed two methods by modifying this method proposed by Furue et al. to make more efficient in [Has21]. Then, the complexities of Hashimoto's techniques on the MQ system with n variables and m equations are estimated as $q^k \cdot MQ(q, m-\alpha-k, m-\alpha) + (m-k) \cdot MQ(q, \alpha, \alpha)$ under the condition $n - m + k \geq \alpha \cdot (m-k)$ and $q^k \cdot (MQ(q, m-\alpha-k, m-\alpha) + MQ(q, \alpha, \alpha)) + (m-\alpha-k) \cdot MQ(q, \alpha, \alpha)$ under the condition $n-m \geq \alpha \cdot (m-k-\alpha)$, where $MQ(q, n, m)$ denotes the complexity of solving the MQ system with n variables and m equations in $\mathbb{F}_q$. In Subsection 7.2, we confirm the security of the proposed parameters by the complexity of the hybrid approach with WXL given by equation (9) using one of these Hashimoto's techniques which has smaller complexity. Note that it is difficult to combine PXL and Hashimoto's techniques since both algorithms utilize the guessed k variables before substituting k values, and thus we apply Thomae and Wolf [TW12] technique to PXL to estimate the complexity in Subsection 7.2.

QR-MQ probelm We finally discuss the security of the QR-MQ problem. In Table 9, for a QR-UOV public key system, we compare the theoretical d_{reg} and experimental d_{reg} using the F4 algorithm. The theoretical d_{reg} is the degree of regularity of F4 as the smallest degree with a non-positive coefficient in $\left(1-z^2\right)^m / (1-z)^{m-k}$, assuming that the system is semi-regular. The experimental d_{reg} is the highest degree among the step degrees, where nonzero polynomials are generated in experiments of F4 using the Magma algebra system [BCP97]. In our experiment, m was set to sufficiently large values so that our computation for one parameter was performed within one day, and v is set equal to $2m$. For the public key of the QR-UOV with $(v+m)$ variables and m equations, we fix the last v variables and execute the hybrid approach by fixing k variables additionally. That is, the direct attack is executed on the system of m equations in $m-k$ variables. As a result, Table 9 shows that the

degrees of regularity obtained experimentally were the same as those obtained theoretically. These results indicate that the difficulty of solving the public key system of QR-UOV is equivalent to that of solving the randomized MQ system.

Remark 5. *In the case of* $(q, v, m, \ell, k) = (7, 30, 15, 3, 1)$ *in Table 9, the experimental* d_{reg} *is larger than the theoretical* d_{reg}*. However, our experiment shows that the experimental* d_{reg} *of the same size randomized quadratic system of* m *equations in* $(m - k)$ *variables over* $\mathbb{F}_7$ *is not different from our experimental* d_{reg} *of* $(q, v, m, \ell, k) = (7, 30, 15, 3, 1)$*.*

8.4 Key Recovery Attacks on UOV

This subsection recalls some proposed existing key recovery attacks, the Kipnis-Shamir [KS98], reconciliation [DYC+08], and intersection [Beu21] attacks. These key recovery attacks can be performed on the following two problems:

- $\mathrm{UOV}(q^\ell, v/\ell, m/\ell, m)$,
- $\mathrm{UOV}(q, v, m, m)$,

where $\mathrm{UOV}(q, v, o, m)$ denotes the plain UOV with v vinegar variables, o oil variables, and m equations over $\mathbb{F}_q$. The first one is corresponding to one of the underlying problems of our security proof obtained by the pull-back transformation descrived in Lemma 3, and the second one is enabled by ignoring the quotient ring structure of QR-UOV. This subsection describes the behavior of the key recovery attacks on $\mathrm{UOV}(q, v, o, m)$, and thus, by substituting $(q^\ell, v/\ell, m/\ell, m)$ and (q, v, m, m) for (q, v, o, m) in the following estimations, we can obtain the complexity of the key recovery attacks on $\mathrm{UOV}(q^\ell, v/\ell, m/\ell, m)$ and $\mathrm{UOV}(q, v, m, m)$, respectively.

Recall that the key recovery attacks aim to obtain the subspace $\mathcal{S}^{-1}(\mathcal{O})$ of $\mathbb{F}_q^n$, where $\mathcal{O}$ is the oil subspace defined as

$$\mathcal{O} := \left\{ (0, \ldots, 0, \alpha_1, \ldots, \alpha_o)^\top \mid \alpha_i \in \mathbb{F}_q \right\}.$$

8.4.1 Kipnis-Shamir Attack

The Kipnis-Shamir attack [KS98] chooses two invertible matrices W_i, W_j from the set of linear combinations of the representation matrices $P_1, \ldots, P_m$ for the public key. Then, it probabilistically recovers a part of the subspace $\mathcal{S}^{-1}(\mathcal{O})$ by computing the invariant subspace of $W_i^{-1} W_j$. The complexity of the Kipnis-Shamir attack is estimated as

$$O\left(q^{v-o-1} \cdot o^4\right).$$

Grover's algorithm [Gro96] can be used to reduce the complexity for a quantum adversary to

$$O\left(q^{\frac{v-o-1}{2}} \cdot o^4\right).$$

Then, the complexity of the Kipnis-Shamir attack for classical and quantum adversaries against UOV$(q^\ell, v/\ell, m/\ell, m)$ is given as

$$O\left(q^{v-m-\ell} \cdot (m/\ell)^4\right), \quad O\left(q^{\frac{v-m-\ell}{2}} \cdot (m/\ell)^4\right).$$

Furthermore, the complexity of the Kipnis-Shamir attack for classical and quantum adversaries against UOV(q, v, m, m) is given as

$$O\left(q^{v-m-1} \cdot m^4\right), \quad O\left(q^{\frac{v-m-1}{2}} \cdot m^4\right).$$

8.4.2 Reconciliation Attack

The reconciliation attack [DYC+08] treats a vector y of $\mathcal{S}^{-1}(\mathcal{O})$ as variables and solves the quadratic system $y^\top P_i y = 0$ $(i \in [m])$. Here, the dimension of $\mathcal{S}^{-1}(\mathcal{O})$ is o, and thus if we impose affine constraints, we then solve a system of m equations in $n-o = v$ variables and still have a solution with high probability. Parameters of UOV are generally set to satisfy $v > m$ for the security against the Kipnis-Shamir attack, and in this case the system of $y^\top P_i y = 0$ has a large number of solutions. Therefore, to determine a solution uniquely, we need to solve the following system to find multiple vectors $y_1, \ldots, y_k$ of $\mathcal{S}^{-1}(\mathcal{O})$:

$$\begin{cases} y_j^\top P_i y_j = 0 & (1 \le i \le m, 1 \le j \le k), \\ y_j^\top P_i y_\ell = 0 & (1 \le i \le m, 1 \le j < \ell \le k). \end{cases}$$

We here lower bound the complexity of solving this problem by that of solving the MQ problem with v variables and equations. On the other hand, if the number v of the vinegar variables is smaller than the number m of equations, then the complexity of the reconciliation attack is estimated as that of solving a quadratic system of m equations in v variables. We estimate the complexity of solving these problems with the complexity of the hybrid approach with WXL in equations (9) and (11).

Then, the complexity agianst UOV$(q^\ell, v/\ell, m/\ell, m)$ is given by

$$\min_k \left(O\left(q^{\ell \cdot k} \cdot 3 \cdot \binom{v/\ell - k + 2}{2} \cdot \binom{d_{reg} + v/\ell - k}{d_{reg}}^2 \right) \right),$$

where $0 \le k \le v/\ell$, since $v/\ell < m$. Furthermore, the complexity agianst UOV(q, v, m, m) is given by

$$\min_k \left(O\left(q^k \cdot 3 \cdot \binom{v - k + 2}{2} \cdot \binom{d_{reg} + v - k}{d_{reg}}^2 \right) \right),$$

where $0 \le k \le v$.

8.4.3 Intersection Attack

In [Beu21], Beullens proposed a new key recovery attack against UOV, called an intersection attack. In the case of $v < 2o$, for an integer $k \geq 2$ satisfying $k < \frac{v}{v-o}$, let $L_1, \ldots, L_k$ be k invertible matrices randomly chosen from a set of linear combinations of the representation matrices $P_1, \ldots, P_m$ for the public key. This attack then solves the following equations for $\mathbf{y} \in \mathbb{F}_q^n$:

$$\begin{cases} (L_j^{-1}\mathbf{y})^\top P_i (L_j^{-1}\mathbf{y}) = 0 & (1 \leq i \leq m, 1 \leq j \leq k), \\ (L_j^{-1}\mathbf{y})^\top P_i (L_\ell^{-1}\mathbf{y}) = 0 & (1 \leq i \leq m, 1 \leq j < \ell \leq k). \end{cases} \tag{12}$$

Note that, for a solution $\mathbf{z}$ for this system, a vector in $\mathcal{S}^{-1}(\mathcal{O})$ is not $\mathbf{z}$ but $L_j^{-1}\mathbf{z}$ unlike the Kipnis-Shamir and reconciliation attacks. The solution space obtained from the above equation has $ko - (k-1)v$ dimensions. Thus, its complexity is equivalent to that of solving the quadratic system with $n - (ko - (k-1)v) = kv - (k-1)o$ variables and $\binom{k+1}{2}m - 2\binom{k}{2}$ equations owing to its linear dependency. The value of k is generally chosen such that the complexity of solving the above system takes the minimum value under the condition of $k < \frac{v}{v-o}$. On the other hand, in the case of $v \geq 2o$, the intersection attack becomes a probabilistic algorithm, which solves the system of equation (12) as $k = 2$ with n variables and $(3m-2)$ equations and one of solutions is a target vector with a probability of approximately $q^{-v+2o-1}$. Therefore, its complexity is estimated by q^{v-2o+1} times the complexity of solving the quadratic system with n variables and $(3m-2)$ equations. We estimate the complexity of solving these problems with the complexity of the hybrid approach with WXL in equations (9) and (11).

Then, the complexity agianst UOV$(q^\ell, v/\ell, m/\ell, m)$ is given by

$$\min_k \left(O\left(q^{v/\ell - 2m/\ell + 1} \cdot q^k \cdot 3 \cdot \binom{n/\ell - k + 2}{2} \cdot \binom{d_{reg} + n/\ell - k}{d_{reg}}^2 \right) \right),$$

where $0 \leq k \leq n/\ell$, since $v/\ell > 2m/\ell$. Furthermore, the complexity agianst UOV(q, v, m, m) is given by

$$\min_k \left(O\left(q^{v-2m+1} \cdot q^k \cdot 3 \cdot \binom{n - k + 2}{2} \cdot \binom{d_{reg} + n - k}{d_{reg}}^2 \right) \right),$$

where $0 \leq k \leq n$, since $v > 2m$.

8.5 Rectangular MinRank Attack

This subsection shows that the rectangular MinRank attack [Beu21] is applicable to only UOV(q^ℓ, V, M, m), where $V = v/\ell$ and $M = m/\ell$, and estimates the complexity.

The rectangular MinRank attack was recently proposed for the Rainbow scheme by Beullens, and it tries to solve a new MinRank problem obtained by

transforming the public key of Rainbow. Rainbow is a multi-layered variant of the UOV scheme, and UOV has resistance to all MinRank attacks since UOV does not have a structure of MinRank problem. However, we show that the rectangular MinRank is applicable to $\mathrm{UOV}(q^\ell, V, M, m)$ which is one of the underlying problems of the security of QR-UOV. Note that we here suppose that $(P_1, \ldots, P_m)$ and $(F_1, \ldots, F_m), S$ are matrices representing the public and secret keys of UOV with parameters (q^ℓ, V, M, m).

Before describing the rectangular MinRank attack, we introduce a way of transforming sets of matrices used in the attack. Let $(G_1, \ldots, G_m)$ be a set of n-by-n matrices over $\mathbb{F}_q$, and $\mathbf{g}_i^{(j)}$ denotes the j-th column vector of G_i, namely,

$$G_i = \begin{pmatrix} \mathbf{g}_i^{(1)} & \mathbf{g}_i^{(2)} & \cdots & \mathbf{g}_i^{(n)} \end{pmatrix} \in M_n(\mathbb{F}_q).$$

Then, we define the new set $(\tilde{G}_1, \ldots, \tilde{G}_n)$ of n-by-m matrices as follows:

$$\begin{aligned} \tilde{G}_1 &:= \begin{pmatrix} \mathbf{g}_1^{(1)} & \mathbf{g}_2^{(1)} & \cdots & \mathbf{g}_m^{(1)} \end{pmatrix}, \\ \tilde{G}_2 &:= \begin{pmatrix} \mathbf{g}_1^{(2)} & \mathbf{g}_2^{(2)} & \cdots & \mathbf{g}_m^{(2)} \end{pmatrix}, \\ &\vdots \\ \tilde{G}_n &:= \begin{pmatrix} \mathbf{g}_1^{(n)} & \mathbf{g}_2^{(n)} & \cdots & \mathbf{g}_m^{(n)} \end{pmatrix}. \end{aligned}$$

Then, when we apply this deformation to $(P_1, \ldots, P_m)$ and $(F_1, \ldots, F_m)$, we have

$$(\tilde{P}_1, \ldots, \tilde{P}_n) = (S^\top \tilde{F}_1, \ldots, S^\top \tilde{F}_n) \cdot S.$$

For the proposed parameters in Subsection 4.4, we have $m > V > M$. From this relation, it is easily seen that the deformation matrices $\tilde{F}_{V+1}, \ldots, \tilde{F}_N \in M_{N\times m}(\mathbb{F}_q)$ are of rank $\leq V$ since they have the following form:

$$\begin{pmatrix} *_{V\times m} \\ 0_{M\times m} \end{pmatrix}.$$

Then, there exists a linear combination of $\tilde{P}_1, \ldots, \tilde{P}_N \in M_{n\times m}(\mathbb{F}_q)$ whose rank is $\leq V$, and thus, as in Rainbow, the rectangular MinRank attack can be applied to $\mathrm{UOV}(q^\ell, V, M, m)$. In order to estimate the complexity, we describe the attack in detail.

The rectangular MinRank attack tries to find a non-zero element of $\mathcal{S}^{-1}(\mathcal{O})$. As in the case of Rainbow, the rectangular MinRank attack against UOV with (q^ℓ, V, M, m) is constructed as follows. Since $\dim\left(\mathcal{S}^{-1}(\mathcal{O})\right) = m$, there exists a non-zero N-by-1 vector with the following form:

$$\mathbf{a} = (a_1, a_2, \ldots, a_{V+1}, 0, \ldots, 0) \in \mathcal{S}^{-1}(\mathcal{O}).$$

Then, it is shown that

$$\sum_{i=1}^{V+1} a_i \tilde{P}_i = (\tilde{P}_1, \ldots, \tilde{P}_N) \cdot \mathbf{a} = (S^\top \tilde{F}_1, \ldots, S^\top \tilde{F}_N) \cdot (S \cdot \mathbf{a})$$

is a linear combination of $S^\top \tilde{F}_{v+1}, \ldots, S^\top \tilde{F}_N$. Thus, this linear combination is of rank $\leq V$. Namely, the vector $\mathbf{a}$ gives a solution to the MinRank problem for $(\tilde{P}_1, \ldots, \tilde{P}_{V+1})$ with the target rank V. Moreover, we have

$$p_1(\mathbf{a}) = \cdots = p_m(\mathbf{a}) = 0.$$

As a result, the vector $\mathbf{a} = (a_1, a_2, \ldots, a_{V+1}, 0, \ldots, 0)$ we want to find is a common solution to the following problems:

$$\text{(i) Rank} \left(\sum_{i=1}^{V+1} a_i \tilde{P}_i \right) \leq V,$$
$$\text{(ii) } p_1(\mathbf{a}) = \cdots = p_m(\mathbf{a}) = 0.$$

Complexity analysis

We then describe the estimation of the complexity to solve the above problems (i) and (ii). This is done along Beullens' estimation [Beu21] for the rectangular MinRank attack against Rainbow. Note that the characteristic of $\mathbb{F}_{q^\ell}$ is always odd in QR-UOV.

First, consider problem (i). Fix an integer m' such that $V + 1 \leq m' \leq m$. Let $\tilde{P}'_i$ be the $N \times m'$ matrix obtained by removing the column vectors from $(m'+1)$-th to m-th of $\tilde{P}_i$. Then one considers to apply the support minor modeling method [BBC+20] to the MinRank problem $(\tilde{P}'_1, \ldots, \tilde{P}'_{V+1})$ with the target rank V. Let I' be the ideal in $\mathbb{F}_{q^\ell}[\mathbf{a}, \mathbf{c}]$ generated by the bilinear equations obtained from the support minor modeling, where $\mathbf{c}$ is the set of $\binom{m'}{V+1}$ minor variables. For $b \in \mathbb{N}$, set

$$R'(b) := \sum_{i=1}^{b} (-1)^{i+1} \binom{m'}{V+i} \binom{N+i-1}{i} \binom{V+1+b-i}{b-i}.$$

Let $I'_{b,1}$ be the subspace of $(b,1)$-degree homogeneous polynomials of I' in $\mathbb{F}_{q^\ell}[\mathbf{a}, \mathbf{c}]$. Then, Bardet et al. [BBC+20] calculated that $\dim_{\mathbb{F}_{q^\ell}} I'_{b,1} = R'(b)$ for $1 \leq b \leq V+1$.

Next, one considers adding problem (ii). We assume that $p_1(\mathbf{a}), \ldots, p_m(\mathbf{a})$ behave as a semi-regular system, where

$$\mathbf{a} = (a_1, a_2, \ldots, a_{V+1}, 0, \ldots, 0).$$

Let I be the ideal generated by I' and $p_1(\mathbf{a}), \ldots, p_m(\mathbf{a})$, namely,

$$I := I' + \langle p_1(\mathbf{a}), \ldots, p_m(\mathbf{a}) \rangle \subset \mathbb{F}_{q^\ell}[\mathbf{a}, \mathbf{c}].$$

Moreover, set

$$G'(t_1, t_2) := \binom{m'}{V} t_2 + \sum_{b=1}^{V+1} \left(\binom{m'}{V} \binom{V+b-1}{b} - R'(b) \right) t_1^b t_2,$$
$$G(t_1, t_2) := G'(t_1, t_2) \cdot (1 - t_1^2)^m.$$

Table 10: Experiments for b_{min} and $b_{min}^{(predict)}$

(q, V, M, l)	m'	$b_{min}^{(predict)}$	b_{min}
$(7, 5, 2, 3)$	6	4	4
$(7, 6, 3, 3)$	7	3	3
	8	3	3
	9	2	2
$(7, 7, 3, 3)$	8	4	4
	9	3	3
$(7, 8, 3, 3)$	9	5	5

Let $b_{min} \in \mathbb{N}$ be the minimum of b such that

$$\dim_{\mathbb{F}_{q^\ell}} I_{b,1} = \dim_{\mathbb{F}_{q^\ell}} \mathbb{F}_{q^\ell}[\mathbf{a}, \mathbf{c}]_{b,1} - 1.$$

Then, following Beullens' estimation, we can state that b_{min} is predicted by

$$b_{min}^{(predict)} := \min\left\{b \mid G(t_1, t_2)_{b,1} \leq 1\right\}, \tag{13}$$

where $G(t_1, t_2)_{b,1}$ is the coefficient of $t_1^b t_2$.

Finally, by applying to $I_{b_{min},1}$ the bilinear XL algorithm with Wiedemann algorithm [Wie86], we can find a solution $\mathbf{a}$ to problem (i) and (ii) with the following complexity:

$$3\binom{m'}{V}^2 \binom{V + b_{min} - 1}{b_{min}}^2 (V+1)^2. \tag{14}$$

In Table 10, we experimented that b_{min} is equal to $b_{min}^{(predict)}$ for some small parameters. As seen in Table 10, we have $b_{min} = b_{min}^{(predict)}$, and thus we use $b_{min}^{(predict)}$ instead of b_{min} to estimate the complexity of the rectangular MinRank attack against QR-UOV theoretically in Subsection 7.2.

8.6 Lifting Method

Lifting method is a method of attacking QR-UOV by diagonalizing the matrices in $\mathcal{A}_f$ over the extension field $\mathbb{F}_{q^\ell}$ and was proposed in [FIKT21]. In this subsection, we show that the lifting method is essentially the same as the pull-back method given in the proof of Lemma 3. To explain it, we prepare some results.

Theorem 6. *With the same notation as in Theorem 5,*

(i) There exists an invertible matrix $L \in \mathbb{F}_{q^\ell}^{\ell \times \ell}$ such that

$$L^{-1}\Phi_x^f L = \begin{pmatrix} x & & & & \\ & x^q & & & \\ & & x^{q^2} & & \\ & & & \ddots & \\ & & & & x^{q^{\ell-1}} \end{pmatrix}.$$

In particular, this L diagonalizes any matrix in $\mathcal{A}_f$.

(ii) The matrix L described in (i) satisfies the condition that $L^\top W L$ is diagonal. Therefore, we can write

$$L^\top W L = \begin{pmatrix} \alpha_0 & & & \\ & \alpha_1 & & \\ & & \ddots & \\ & & & \alpha_{\ell-1} \end{pmatrix}.$$

See [FIKT21] for the proof of this theorem. We recall the idea of the lifting method stated in [FIKT21]. The first and second statements in the theorem show that for any $g \in \mathbb{F}_q[x]/(f) \cong \mathbb{F}_{q^\ell}$ the matrix $L^\top W \Phi_g^f L$ is diagonal. This indicates that $P_1, \ldots, P_m$ of QR-UOV can be transformed into block diagonal matrices for which the block size is $N \times N$. Let $L^{(N)} = I_N \otimes L$ be an $n \times n$ block diagonal matrix with block size ℓ ($n = \ell \cdot N$), for which the N diagonal blocks are L. Then, $(L^{(N)})^\top P_i L^{(N)}$ ($i \in [m]$) become block matrices wherein every component is in a diagonal form. Furthermore, there exists a permutation matrix A such that $(L^{(N)}A)^\top P_i (L^{(N)}A)$ is a block diagonal matrix with block size N, and let $\bar{L} := L^{(N)}A$. The transformed matrices $\bar{L}^\top P_i \bar{L}$ can be represented by $(\bar{L}^{-1}S\bar{L})^\top (\bar{L}^\top F_i \bar{L})\,(\bar{L}^{-1}S\bar{L})$. Then, $\bar{L}^\top F_i \bar{L}$ is the diagonal concatenation of ℓ smaller matrices, similar to $\bar{L}^\top P_i \bar{L}$. Furthermore, $\bar{L}^{-1}S\bar{L}$ is also the diagonal concatenation of ℓ smaller matrices from (i) in Theorem 6. Then, owing to the structure of F_i, every diagonal block of $\bar{L}^\top F_i \bar{L}$ has an $M \times M$ zero block, similar to F_i. Therefore, each diagonal block of $\bar{L}^\top P_i \bar{L}$ has the same form as the matrix representing the public key of UOV with V vinegar variables and M oil variables over $\mathbb{F}_{q^\ell}$. The lifting method proposed in [FIKT21] executes the key recovery attacks on one of such diagonal blocks.

In the following, we describe such diagonal blocks in detail and show that the lifting method is essentially the same as the pull-back method. Let $A \in \mathbb{F}_q^{n \times n}$ be a permutation matrix such that

$$A^\top (X \otimes Y)\, A = Y \otimes X$$

for any $X \in \mathbb{F}_q^{N \times N}$ and $Y \in \mathbb{F}_q^{\ell \times \ell}$. Also, we recall the equation in Lemma 3

$$P_k = \sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} \otimes W \Phi_{x^i}^f.$$

Then the transformation in the above lifting method is described as follows:

$$
\begin{aligned}
&A^\top (L^{(N)})^\top P_k L^{(N)} A = A^\top (L^{(N)})^\top \left(\sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} \otimes W \Phi^f_{x^i} \right) L^{(N)} A \\
=&A^\top \left(\sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} \otimes L^\top W \Phi^f_{x^i} L \right) A = A^\top \left(\sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} \otimes L^\top W L \cdot L^{-1} \Phi^f_{x^i} L \right) A \\
=&A^\top \left(\sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} \otimes \begin{pmatrix} \alpha_0 x^i & & & & \\ & \alpha_1 x^{qi} & & & \\ & & \alpha_2 x^{q^2 i} & & \\ & & & \ddots & \\ & & & & \alpha_{\ell-1} x^{q^{\ell-1} i} \end{pmatrix} \right) A \\
=&\sum_{i=0}^{\ell-1} \begin{pmatrix} \alpha_0 x^i & & & & \\ & \alpha_1 x^{qi} & & & \\ & & \alpha_2 x^{q^2 i} & & \\ & & & \ddots & \\ & & & & \alpha_{\ell-1} x^{q^{\ell-1} i} \end{pmatrix} \otimes \bar{P}_k^{(i)} \\
=&\begin{pmatrix} \alpha_0 \sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} x^i & & & & \\ & \alpha_1 \sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} x^{qi} & & & \\ & & \alpha_2 \sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} x^{q^2 i} & & \\ & & & \ddots & \\ & & & & \alpha_{\ell-1} \sum_{i=0}^{\ell-1} \bar{P}_k^{(i)} x^{q^{\ell-1} i} \end{pmatrix} \\
=&\begin{pmatrix} \alpha_0 \bar{P}_k & & & & \\ & \alpha_1 \bar{P}_{k,q} & & & \\ & & \alpha_2 \bar{P}_{k,q^2} & & \\ & & & \ddots & \\ & & & & \alpha_{\ell-1} \bar{P}_{k,q^{\ell-1}} \end{pmatrix}.
\end{aligned}
$$

Here, we have set $\bar{P}_{k,q^a} := (p_{i,j}^{q^a})_{i,j}$, where $\bar{P}_k = (p_{i,j})$. Therefore, $\bar{P}_{k,q}, \ldots, \bar{P}_{k,q^{\ell-1}}$ are easily recovered from $\bar{P}_k$. Thus, when we consider a key recovery attack using the lifting method, it is enough to treat only $\bar{P}_k$ ($k \in [m]$). Since the pull-back method is also to execute a key recovery attack on $\bar{P}_k$, we conclude that a key recovery attack using the pull-back method is the same as that using the lifting method. The only difference from the pull-back method is that we can apply the direct attack on the system of $\bar{L}^\top P_i \bar{L}$ ($i = 1, \ldots, m$) obtained by applying the lifting method. However, for most cases, this lifting direct attack is not more efficient than the plain direct attack, since the large finite field $\mathbb{F}_{q^\ell}$ disturbs guessing some variables in the hybrid approach. Therefore, we list only the complexity of the plain direct attack in Subsection 7.2.

9 Advantages and Limitations

The main advantages of QR-UOV are

- **Public key and signature size**. The plain UOV is known as a scheme with a small signature and a large public key. (Currently proposed parameters in security level I have approximately 50KB public key and 100B signature [BCH+23].) Our proposed parameter sets with $q = 31$ and $\ell = 3$ reduce the public key size by approximately 50-60% compared with the plain UOV (approximately 20KB in security level I) with a few hundred bits of signature.
- **Efficiency**. The signature generation and verification processes consist of simple linear algebra operations over small finite fields (e.g. $\mathbb{F}_7$, $\mathbb{F}_{31}$, and $\mathbb{F}_{127}$) and thus QR-UOV can be implemented very efficiently.
- **Security**. The EUF-CMA security of QR-UOV is formally proven in the QROM assuming the difficulty of two problems, the UOV and QR-MQ problems. The security of the plain UOV is based on the UOV problem and thus it seems relatively well understood. By contrast, the QR-MQ problem is a new assumption generated by us to construct our security proof. There exists no formal reduction from the QR-MQ problem into the plain MQ problem, but we provide some experimental facts which indicate that the difficulty of solving the QR-MQ problem is equivalent to that of solving the plain MQ problem.
- **Simplicity.** The design of the plain UOV is extremely simple, and our QR-UOV is a natural extension of UOV utilizing the quotient rings structure. We can consider that the research undertaken to obtain from UOV to QR-UOV corresponds to that obtained from LWE to MLWE, where MLWE problem is a generalization of LWE using a module comprising vectors over a ring. Therefore, it requires only a minimum knowledge of algebra to understand and implement the scheme.

The main disadvantage of QR-UOV is the large size of the public key compared with other post-quantum signature schemes such as lattice-based signatures. As we mentioned above, the public key size of QR-UOV is reduced from that of the plain UOV and third round parameters of Rainow. However, some selected lattice-based signature schemes have further small public keys with approximately 1000B in security level I. This disadvantage might make it difficult to apply QR-UOV to constrained devices such as smart cards. However, the increase in memory capabilities in the future will relax the impact of this disadvantage.

References

[AASA+19] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta,

et al. Status report on the first round of the nist post-quantum cryptography standardization process. *US Department of Commerce, NIST*, 2019.

[Bar04] M. Bardet. *Étude des systèms algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.* PhD thesis, Université Pierre et Marie Curie-Paris VI, 2004.

[BBC+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems. In *Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26*, pages 507–536. Springer, 2020.

[BCC+22] Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias Kannwischer. MAYO specification. `https://pqmayo.org/assets/specs/mayo.pdf`, 2022.

[BCH+23] Ward Beullens, Ming-Shing Chen, Shih-Hao Hung, Matthias J. Kannwischer, Bo-Yuan Peng, Cheng-Jhih Shih, and Bo-Yin Yang. Oil and vinegar: Modern parameters and implementations. Cryptology ePrint Archive, Paper 2023/059, 2023. `https://eprint.iacr.org/2023/059`.

[BCP97] Wieb Bosma, John Cannon, and Catherine Playoust. The Magma algebra system. I. The user language. *J. Symbolic Comput.*, 24(3-4):235–265, 1997. Computational algebra and number theory (London, 1993).

[Beu21] Ward Beullens. Improved cryptanalysis of UOV and Rainbow. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 348–373, Cham, 2021. Springer International Publishing.

[Beu22] Ward Beullens. Breaking Rainbow takes a weekend on a laptop. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 464–479, Cham, 2022. Springer Nature Switzerland.

[BFP09] Luk Bettale, Jean-Charles Faugere, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2009.

[BFS03] Magali Bardet, Jean-Charles Faugere, and Bruno Salvy. *Complexity of Gröbner basis computation for Semi-regular Overdetermined sequences over F_2 with solutions in F_2.* PhD thesis, INRIA, 2003.

[BFSY05] Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Bo-Yin Yang. Asymptotic behavior of the index of regularity of quadratic semi-regular polynomial systems. In *8th International Symposium on Effective Methods in Algebraic Geometry*, 2005.

[BGV14] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory (TOCT)*, 6(3):1–36, 2014.

[CDP23] Sanjit Chatterjee, M Prem Laxman Das, and Tapas Pandit. Revisiting the security of salted UOV signature. In *Progress in Cryptology–INDOCRYPT 2022: 23rd International Conference on Cryptology in India, Kolkata, India, December 11–14, 2022, Proceedings*, pages 697–719. Springer, 2023.

[CHT12] Peter Czypek, Stefan Heyse, and Enrico Thomae. Efficient implementations of MQPKS on constrained devices. In Emmanuel Prouff and Patrick Schaumont, editors, *Cryptographic Hardware and Embedded Systems – CHES 2012*, pages 374–389, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[CKPS00] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, pages 392–407, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[CKY21] Tung Chou, Matthias J. Kannwischer, and Bo-Yin Yang. Rainbow on cortex-m4. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):650–675, 2021.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 238–268, 2018.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 164–175, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[DYC+08] Jintai Ding, Bo-Yin Yang, Chia-Hsin Owen Chen, Ming-Shing Chen, and Chen-Mou Cheng. New differential-algebraic attacks and reparametrization of Rainbow. In Steven M. Bellovin, Rosario Gennaro, Angelos Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security*, pages 242–257, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Fau99] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999.

[Fau02] J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In *ISSAC 2002*, pages 75–83. ACM, 2002.

[FI23] Hiroki Furue and Yasuhiko Ikematsu. A minrank attack against variants of UOV signature scheme. In *Symposium on Cryptography and Information Security (SCIS), 1A1-2*, 2023.

[FIH+23] Hiroki Furue, Yasuhiko Ikematsu, Fumitaka Hoshino, Yutaro Kiyomura, Tsunekazu Saito, and Tsuyoshi Takagi. Secure parameters for multivariate polynomial signature scheme QR-UOV. In *Symposium on Cryptography and Information Security (SCIS), 1A1-4*, 2023. `http://crypto.mist.i.u-tokyo.ac.jp/publications/1A1-4.pdf`.

[FIKT21] Hiroki Furue, Yasuhiko Ikematsu, Yutaro Kiyomura, and Tsuyoshi Takagi. A new variant of unbalanced oil and vinegar using quotient ring: QR-UOV. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 187–217, Cham, 2021. Springer International Publishing.

[FK21] Hiroki Furue and Momonari Kudo. Polynomial XL: A variant of the XL algorithm using Macaulay matrices over polynomial rings. Cryptology ePrint Archive, Paper 2021/1609, 2021. `https://eprint.iacr.org/2021/1609`.

[FKI+20] Hiroki Furue, Koha Kinjo, Yasuhiko Ikematsu, Yacheng Wang, and Tsuyoshi Takagi. A structural attack on block-anti-circulant UOV at SAC 2019. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography*, pages 323–339, Cham, 2020. Springer International Publishing.

[FNT21] Hiroki Furue, Shuhei Nakamura, and Tsuyoshi Takagi. Improving Thomae-Wolf algorithm for solving underdetermined multivariate quadratic polynomial problem. In Jung Hee Cheon and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography*, pages 65–78, Cham, 2021. Springer International Publishing.

[FT23] Hiroki Furue and Tsuyoshi Takagi. Fast enumeration algorithm for multivariate polynomials over general finite fields. Cryptology ePrint Archive, Paper 2023/619, 2023. `https://eprint.iacr.org/2023/619`.

[GJ90] Michael R. Garey and David S. Johnson. *Computers and intractability; a guide to the theory of NP-completeness*. W. H. Freeman & Co., USA, 1990.

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. Association for Computing Machinery.

[Has21] Yasufumi Hashimoto. Minor improvements of algorithm to solve under-defined systems of multivariate quadratic equations. Cryptology ePrint Archive, Paper 2021/1045, 2021. `https://eprint.iacr.org/2021/1045`.

[Has22] Yasufumi Hashimoto. An elementary construction of QR-UOV. Cryptology ePrint Archive, Paper 2022/145, 2022. `https://eprint.iacr.org/2022/145`.

[HFI+23] Fumitaka Hoshino, Hiroki Furue, Yasuhiko Ikematsu, Tsunekazu Saito, Yutaro Kiyomura, and Tsuyoshi Takagi. Efficient software implementation of signature scheme QR-UOV. In *Symposium on Cryptography and Information Security (SCIS), 1A1-5*, 2023. `http://crypto.mist.i.u-tokyo.ac.jp/publications/1A1-5.pdf`.

[JS19] Samuel Jaques and John M. Schanck. Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In *Advances in Cryptology – CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part I*, page 32–61, Berlin, Heidelberg, 2019. Springer-Verlag.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 206–222, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[KS98] Aviad Kipnis and Adi Shamir. Cryptanalysis of the oil and vinegar signature scheme. In Hugo Krawczyk, editor, *Advances in Cryptology — CRYPTO '98*, pages 257–266, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[KX22] Haruhisa Kosuge and Keita Xagawa. Probabilistic hash-and-sign with retry in the quantum random oracle model. Cryptology ePrint Archive, Paper 2022/1359, 2022. `https://eprint.iacr.org/2022/1359`.

[MKJR16] Kathleen Moriarty, Burt Kaliski, Jakob Jonsson, and Andreas Rusch. PKCS #1: RSA cryptography specifications version 2.2. Technical report, Internet Engineering Task Force, 2016.

[NIS] NIST: Post-quantum cryptography CSRC. `https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization`.

[NIS22] NIST: Call for additional digital signature schemes for the post-quantum cryptography standardization process. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf, 2022.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[Sho01] Victor Shoup. A proposal for an ISO standard for public key encryption. Cryptology ePrint Archive, Paper 2001/112, 2001. https://eprint.iacr.org/2001/112.

[SP20] Alan Szepieniec and Bart Preneel. Block-anti-circulant unbalanced oil and vinegar. In Kenneth G. Paterson and Douglas Stebila, editors, *Selected Areas in Cryptography – SAC 2019*, pages 574–588, Cham, 2020. Springer International Publishing.

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. On provable security of UOV and HFE signature schemes against chosen-message attack. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 68–82, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[TW12] Enrico Thomae and Christopher Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography – PKC 2012*, pages 156–171, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[Wie86] Douglas Wiedemann. Solving sparse linear equations over finite fields. *IEEE transactions on information theory*, 32(1):54–62, 1986.

[YC05] Bo-Yin Yang and Jiun-Ming Chen. All in the XL family: Theory and practice. In Choon-sik Park and Seongtaek Chee, editors, *Information Security and Cryptology – ICISC 2004*, pages 67–86, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[YCBC07] Bo-Yin Yang, Owen Chia-Hsin Chen, Daniel J. Bernstein, and Jiun-Ming Chen. Analysis of QUAD. In Alex Biryukov, editor, *Fast Software Encryption*, pages 290–308, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.