QR-UOV without Rejection Sampling: Security Analysis and High-Speed Implementation

Hiroshi Amagasa*[1], Hiroki Furue[2], Rei Ueno[3,1], and Naofumi Homma[1]

[1]Tohoku University, Japan
[2]NTT Social Informatics Laboratories, Japan
[3]Kyoto University, Japan

Abstract

QR-UOV is a multivariate signature scheme derived from UOV that achieves compact public keys by exploiting quotient-ring structure, making it a promising candidate for post-quantum digital signatures. In QR-UOV, most parts of the public map are derived from the public key seed using a PRG. This public key expansion for QR-UOV includes rejection sampling to generate coefficients uniformly over $\mathbb{F}_q$, since QR-UOV uses a small odd-prime base field. However, this rejection sampling introduces extra data movement and irregular control flow. For the recommended parameter set, public-key expansion accounts for nearly 90% of the QR-UOV verification time.

In this paper, we propose No Rejection Sampling (NoRS) QR-UOV, a variant of QR-UOV with rejection-sampling-free public-key expansion, which leaves the generation of secret-dependent coefficients unchanged. Concretely, the rejected value q is deterministically mapped to 0, which simplifies coefficient generation but introduces a slight bias in the resulting coefficient distribution. We evaluate the security impact of this modification through both theoretical and concrete analyses. Our results suggest that, for the proposed parameter sets, NoRS QR-UOV maintains the claimed security levels against the considered attacks.

On the implementation side, we develop a high-speed implementation of NoRS QR-UOV for x86 processors with AES-NI and AVX2. Benchmark results on a Skylake platform show that NoRS consistently accelerates QR-UOV at all security levels, with the largest gain in signature verification. Under the standard Round-2 parameter settings, benchmark results show that NoRS QR-UOV reduces the verification cost from 0.43 to 0.30 Mcycles at security level I, with comparable improvements at security levels III and V, corresponding to about 1.4× speedup over standard QR-UOV. Overall, the results suggest that relaxing coefficient uniformity in public-key expansion is a practical and effective design choice for QR-UOV.

Keywords: post-quantum cryptography, multivariate public-key cryptography, QR-UOV, rejection sampling, AES-NI

1 Introduction

Practical deployment of post-quantum cryptography calls for not only security evaluation but also implementation efficiency. This requirement is particularly important for digital signatures, where verification throughput can be a decisive factor in real-world applications. This viewpoint is also reflected in the ongoing NIST standardization process for additional digital signatures, which aims to diversify the standardized portfolio beyond structured-lattice schemes and has explicitly highlighted interest in schemes with short signatures and fast verification.

*hiroshi.amagasa.q7@dc.tohoku.ac.jp

QR-UOV [1] is a multivariate signature scheme derived from UOV [2]. It is one of the candidates selected for Round 2 of the NIST standardization process for additional digital signatures.[1] The main feature of QR-UOV is its quotient-ring structure, which mitigates a major drawback of UOV, namely its large public key. By exploiting the quotient-ring structure, QR-UOV reduces the public-key size by about 50% compared with UOV, while maintaining security. NIST's first-round status report [3] notes that, although QR-UOV is slower than UOV, its performance is still competitive. The report also states that its quotient-ring structure requires further study, and that its implementation leaves room for future optimization.

A previous implementation study [4] improved QR-UOV performance, and also showed that QR-UOV verification is dominated by public-key expansion. For a recommended parameter set $(q, \ell) = (127, 3)$, nearly 90% of the total verification time comes from public-key expansion, with AES-CTR and rejection sampling accounting for 61% and 27%, respectively. Hence, coefficient generation during expansion is a major implementation bottleneck of QR-UOV. This bottleneck closely relates to a distinctive algebraic feature of QR-UOV. Unlike other multivariate candidates in the NIST standardization process for additional digital signatures—UOV [5], SNOVA [6], and MAYO [7]—all of which use characteristic-two fields, QR-UOV uses a small odd-prime base field. As a result, uniform coefficient generation over $\mathbb{F}_q$ in public-key expansion requires rejection sampling. This scheme-specific step introduces extra data movement and irregular control flow, and is a major source of software overhead in verification latency.

In this work, to address this bottleneck, we propose No Rejection Sampling (NoRS) QR-UOV, a variant that removes rejection sampling from public-key expansion while leaving secret-dependent coefficient generation unchanged. Concretely, the rejected value in standard QR-UOV is deterministically mapped to 0. This modification simplifies and regularizes coefficient generation, but it introduces a slight bias in the coefficient distribution. Hence, NoRS QR-UOV must be evaluated from both security and implementation viewpoints. This paper makes three main contributions spanning security analysis, implementation, and performance evaluation.

- We analyze the coefficient bias introduced by NoRS and evaluate its security impact through both theoretical and concrete security analyses. In particular, our analysis suggests that the proposed parameter sets preserve the claimed security levels from the original QR-UOV.
- We develop a high-speed implementation of NoRS QR-UOV on x86 processors with AES-NI and AVX2, including a fused expansion-and-dot-product routine for verification.
- We demonstrate that NoRS consistently accelerates QR-UOV at all security levels, with the largest gain in signature verification.

2 Preliminaries

2.1 UOV and QR-UOV

Multivariate public-key cryptography is based on the hardness of solving systems of multivariate quadratic equations over a finite field. In UOV-type signature schemes, the private key consists of a structured central quadratic map $\mathcal{F} = (f_1, \ldots, f_m) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and a random invertible linear map $\mathcal{S} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ while the public key is their composition $\mathcal{P} = \mathcal{F} \circ \mathcal{S}$. A signature $\boldsymbol{s} \in \mathbb{F}_q^n$ satisfies $\mathcal{P}(\boldsymbol{s}) = \boldsymbol{t}$, where $\boldsymbol{t} \in \mathbb{F}_q^m$ is the vector derived from the message digest and salt. Verification checks this equality.

In UOV [2], the variables are divided into v vinegar variables and m oil variables, with $n = v + m$. Each quadratic polynomial f_i can be represented by an $n \times n$ matrix F_i of the form

$$F_i = \begin{pmatrix} *_{v\times v} & *_{v\times m} \\ *_{m\times v} & 0_{m\times m} \end{pmatrix},$$

[1] In May 2026, NIST selected QR-UOV as a Round 3 candidate in the same standardization process.

where the zero lower-right submatrix means that no products of two oil variables appear in f_i. Hence, once the vinegar variables are fixed, the resulting equations are linear in the oil variables. Let S denote the matrix representation of $\mathcal{S}$. Then the corresponding public-key matrices are given by $P_i = S^\top F_i S$.

QR-UOV [1] is a structured variant of UOV that reduces the public-key size by introducing quotient-ring structure. Let $\mathbb{F}_{q^\ell} = \mathbb{F}_q[x]/(f)$ for an irreducible polynomial $f \in \mathbb{F}_q[x]$ of degree ℓ. For each $g \in \mathbb{F}_{q^\ell}$, define an $\ell \times \ell$ matrix $\Phi_g^f \in \mathbb{F}_q^{\ell\times\ell}$ by

$$(1, x, \ldots, x^{\ell-1})\Phi_g^f = (g, xg, \ldots, x^{\ell-1}g) \pmod f.$$

This gives an injective ring homomorphism from $\mathbb{F}_{q^\ell}$ into $\mathbb{F}_q^{\ell\times\ell}$, so that each structured $\ell \times \ell$ block is determined by only ℓ coefficients over $\mathbb{F}_q$, rather than by ℓ^2 independent matrix entries. In QR-UOV, the public key matrices P_i with $i \in [m]$ are constructed as block matrices whose every component is given as $W\Phi_g^f$ with $g \in \mathbb{F}_{q^\ell}$, where $W \in \mathbb{F}_q^{\ell\times\ell}$ satisfies the condition that $W\Phi_g^f$ is symmetric for any $g \in \mathbb{F}_{q^\ell}$. As a result, QR-UOV achieves a more compact public-key representation than UOV. Under the vinegar–oil partition, each public-key matrix $P_i \in \mathbb{F}_q^{n\times n}$ is partitioned as

$$P_i = \begin{pmatrix} P_{i,1} & P_{i,2} \\ P_{i,2}^\top & P_{i,3} \end{pmatrix}, \tag{1}$$

where $P_{i,1} \in \mathbb{F}_q^{v\times v}$, $P_{i,2} \in \mathbb{F}_q^{v\times m}$, $P_{i,3} \in \mathbb{F}_q^{m\times m}$. Since P_i is symmetric, both $P_{i,1}$ and $P_{i,3}$ are also symmetric. The compressed public key consists of the matrices $P_{i,3}$ and a public seed. The remaining matrices $P_{i,1}$ and $P_{i,2}$ are expanded from the seed by $\mathsf{Expand}_{\mathsf{pk}}$, which uses a PRG instantiated by SHAKE or AES-CTR.

2.2 QR-UOV verification procedure

To prepare for the verification optimizations described later, we briefly review the QR-UOV verification procedure and fix the notation used throughout the paper. We denote by $\boldsymbol{p}_{i,1}$, $\boldsymbol{p}_{i,2}$, and $\boldsymbol{p}_{i,3}$ the compact coefficient-vector representations of the structured matrices $P_{i,1}$, $P_{i,2}$, and $P_{i,3}$, respectively.

For a signature vector $\boldsymbol{s}$, the public map evaluation $\mathcal{P}(\boldsymbol{s}) = \boldsymbol{t}$ can be written in a component-wise manner as

$$t_i = \boldsymbol{s}^\top P_i \boldsymbol{s} \qquad (i \in [m]).$$

Following the previous QR-UOV implementation study [4], each coordinate evaluation can be expressed as a sum of dot products between vectors precomputed from the signatures and vectors regenerated from the compressed public key. Specifically, after computing $(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3)$ from $\boldsymbol{s}$, the public-key expansion expands $(\boldsymbol{p}_{i,1}, \boldsymbol{p}_{i,2})$ from $\mathsf{seed}_{\mathsf{pk}}$ and evaluates

$$t_i' = \boldsymbol{u}_1 \cdot \boldsymbol{p}_{i,1} + \boldsymbol{u}_2 \cdot \boldsymbol{p}_{i,2} + \boldsymbol{u}_3 \cdot \boldsymbol{p}_{i,3},$$

which is equivalent to computing $\boldsymbol{s}^\top P_i \boldsymbol{s}$. Hence, QR-UOV verification can be viewed as public-key expansion followed by dot-product evaluation. Algorithm 1 shows how this view is realized in the verification procedure.

2.3 Rejection sampling in public-key expansion

In QR-UOV, the base-field parameter q is chosen from $\{7, 31, 127\}$, so $q = 2^k - 1$ for $k \in \{3, 5, 7\}$. This Mersenne-prime form is convenient for arithmetic over $\mathbb{F}_q$, but it also affects coefficient generation in public-key expansion. To generate coefficients uniformly over $\mathbb{F}_q$ from pseudorandom bits, the public-key expansion first applies a bitmask to obtain k-bit values in $\{0, \ldots, 2^k - 1\} = \{0, \ldots, q\}$. Since the valid coefficient range is $\{0, \ldots, q - 1\}$, the unique out-of-range value q must be rejected. Thus, standard QR-UOV uses rejection sampling during public-key expansion to obtain a uniform distribution over $\mathbb{F}_q$.

Algorithm 1 Verify(msg, pk, σ)

Input: message msg $\in \{0,\dots,255\}^*$, public key pk encoding $(\mathsf{seed}_{\mathsf{pk}}, \{\boldsymbol{p}_{i,3}\}_{i\in[m]})$, and signature $\sigma \in \{0,1\}^{\lambda} \times \mathbb{F}_q^n$

Output: accept or reject

1: $(\mathsf{seed}_{\mathsf{pk}}, \{\boldsymbol{p}_{i,3}\}_{i\in[m]}) \leftarrow \mathsf{pk}$
2: $(r, \boldsymbol{s}) \leftarrow \sigma$
3: $(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3) \leftarrow \mathsf{ComputeCoefficients}(\boldsymbol{s})$
4: $\mu \leftarrow \mathsf{SHAKE256}(\mathsf{seed}_{\mathsf{pk}} \parallel \mathsf{BytesToBits}(\mathsf{msg}), 512)$
5: $\boldsymbol{t} \leftarrow \mathsf{Hash}(\mu, r)$ ▷ $\boldsymbol{t}, \boldsymbol{t}' \in \mathbb{F}_q^m$
6: **for** i from 1 to m **do**
7: $\quad(\boldsymbol{p}_{i,1}, \boldsymbol{p}_{i,2}) \leftarrow \mathsf{Expand}_{\mathsf{pk}}(\mathsf{seed}_{\mathsf{pk}}, i)$
8: $\quad t'_i \leftarrow \boldsymbol{u}_1 \cdot \boldsymbol{p}_{i,1} + \boldsymbol{u}_2 \cdot \boldsymbol{p}_{i,2} + \boldsymbol{u}_3 \cdot \boldsymbol{p}_{i,3}$ ▷ $t'_i \leftarrow \boldsymbol{s}^{\top} P_i \boldsymbol{s}$
9: **end for**
10: **return** accept if $\boldsymbol{t} = \boldsymbol{t}'$ and reject otherwise.

In the Round-2 QR-UOV specification [8], rejection sampling is realized by overwriting each occurrence of q in the prefix with a non-q value from the remaining sequence. Although the rejection probability is only $1/2^k$, this replacement step still introduces irregular control flow and makes SIMD parallelization less straightforward.

3 Proposed scheme: NoRS QR-UOV

Our proposed scheme follows the QR-UOV Round-2 specification [8] and modifies only the rejection-sampling step in public-key expansion from the public seed $\mathsf{seed}_{\mathsf{pk}}$. All secret-dependent coefficient generation is left unchanged.

Let $q = 2^k - 1$, and let $r \in \{0,\dots,q\}$ be the k-bit value generated from the PRG output during public-key expansion. Instead of rejecting the case $r = q$, NoRS maps the output coefficient $a \in \mathbb{F}_q$ by

$$a := r \bmod q.$$

Equivalently,

$$a = \begin{cases} 0 & \text{if } r = q, \\ r & \text{otherwise.} \end{cases}$$

At the implementation level, the exceptional value $r = q$ can be kept unchanged as a non-canonical representative of $0 \in \mathbb{F}_q$. This modification eliminates the rejection step and induces a slight bias in the distribution of the generated coefficients. The resulting distribution is

$$\Pr[a = 0] = \frac{2}{2^k}, \quad \Pr[a = x] = \frac{1}{2^k} \ (x \in \mathbb{F}_q, x \neq 0).$$

For the recommended parameter set $q = 127$ (i.e., $k = 7$), the only deviation from uniformity is that 0 occurs twice among the 128 possible 7-bit candidates.

In the remainder of this paper, we use the AES-CTR instantiation of $\mathsf{Expand}_{\mathsf{pk}}$ and the recommended Round-2 QR-UOV instances. For each security level, we consider the recommended instance with $q = 127$ and $\ell = 3$ listed below.

- $(q, v, m, \ell) = (127, 156, 54, 3)$ for security level I,
- $(q, v, m, \ell) = (127, 228, 78, 3)$ for security level III,
- $(q, v, m, \ell) = (127, 306, 105, 3)$ for security level V.

4 Security evaluation

This section analyzes the security of the proposed NoRS QR-UOV presented in Section 3. We first provide a security proof for NoRS QR-UOV and then discuss the hardness of the underlying problems.

4.1 Security proof

This subsection provides a proof for the EUF-CMA security of the proposed NoRS QR-UOV. Recall that, in the specification [8], the security of QR-UOV is reduced to the following two problems: distinguishing a UOV public map from a random MQ map, and solving the MQ problem with the QR structure. For the security proof of the proposed scheme, we introduce variants of the underlying problems that incorporate NoRS in the following.

For $q = 2^k - 1$ with some positive integer k, we denote by $D_{\mathrm{NoRS}}(q)$ the NoRS distribution over $\mathbb{F}_q$ defined in Section 3. For $\mathbb{F}_{q^\ell}$ with $\ell \in \mathbb{N}$, we also define $D_{\mathrm{NoRS}}(q^\ell)$ with respect to the fixed polynomial basis $1, x, \ldots, x^{\ell-1}$ over $\mathbb{F}_q$ as the distribution of

$$g = \sum_{j=0}^{\ell-1} g_j x^j \in \mathbb{F}_{q^\ell},$$

where each base-field coefficient g_j is sampled independently from $D_{\mathrm{NoRS}}(q)$.

- **NoRS UOV problem.** For parameters (q, v, o, m), consider the public key map $\mathcal{P}$ of the NoRS variant of UOV with v vinegar variables, o oil variables, and m equations over $\mathbb{F}_q$. More specifically, we let S' be a randomly chosen matrix in $\mathbb{F}_q^{v\times o}$. For each $i \in [m]$, we take a symmetric matrix $P_{i,1} \leftarrow D_{\mathrm{NoRS}}(q)^{v\times v}$ and $P_{i,2} \leftarrow D_{\mathrm{NoRS}}(q)^{v\times o}$ and set $P_{i,3} \in \mathbb{F}_q^{o\times o}$ as

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

  Then, we define P_i as in (1) and $\mathcal{P}(\mathbf{x}) = (\mathbf{x}^\top P_1 \mathbf{x}, \ldots, \mathbf{x}^\top P_m \mathbf{x})$.

  By contrast, we construct an MQ map $\mathcal{P}'$ with $v + o$ variables and m equations over $\mathbb{F}_q$ without the UOV structure. More specifically, for each $i \in [m]$, the blocks $P_{i,1}$ and $P_{i,2}$ are sampled in the same way as in $\mathcal{P}$. The difference from the case of $\mathcal{P}$ is that $P_{i,3}$ is taken as

  $$P_{i,3} = S_i'^{\top} P'_{i,1} S'_i - {P'_{i,2}}^{\top} S'_i - S_i'^{\top} P'_{i,2},$$

  where S'_i is sampled independently from $\mathbb{F}_q^{v\times o}$, and $P'_{i,1}$ and $P'_{i,2}$ are sampled independently in the same way as $P_{i,1}$ and $P_{i,2}$, respectively. This construction makes the distributions of the corresponding blocks $P_{i,1}$, $P_{i,2}$, and $P_{i,3}$ match those in $\mathcal{P}$, while removing the structure of the secret subspace of UOV.

  Then, the NoRS UOV problem asks to distinguish these two MQ maps $\mathcal{P}$ and $\mathcal{P}'$.

- **NoRS QR-MQ problem.** For parameters (q, v, o, m, ℓ), let $V = v/\ell$, $O = o/\ell$, and $n = v + o$. Consider an MQ map over $\mathbb{F}_{q^\ell}$ constructed in the same way as $\mathcal{P}'(\mathbf{x}) = (\mathbf{x}^\top P_1 \mathbf{x}, \ldots, \mathbf{x}^\top P_m \mathbf{x})$ in the NoRS UOV problem with parameters (q^ℓ, V, O, m).

  For each $i \in [m]$, we obtain $\bar{P}_i \in \mathbb{F}_q^{n\times n}$ by expanding each element $g \in \mathbb{F}_{q^\ell}$ of P_i to an $\ell \times \ell$ block $W\Phi_g^f$ over $\mathbb{F}_q$. We then define an MQ map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ such that

  $$\mathcal{P}(\mathbf{x}) = \left(\mathbf{x}^\top \bar{P}_1 \mathbf{x}, \ldots, \mathbf{x}^\top \bar{P}_m \mathbf{x}\right)^\top.$$

  Then, the NoRS QR-MQ problem asks to solve $\mathcal{P}(\mathbf{x}) = \mathbf{t}$ for a randomly chosen $\mathbf{t} \in \mathbb{F}_q^m$.

Then, we can reduce the EUF-CMA security of the NoRS QR-UOV to the above two problems as follows:

Theorem 1 (EUF-CMA security of NoRS QR-UOV)**.** *For any quantum EUF-CMA adversary* $\mathcal{A}_{\mathsf{cma}}$ *against the NoRS QR-UOV with parameters* (q, v, m, ℓ) *issuing at most* q_{sign} *classical queries to the signing oracle and* q_{qro} *(quantum) random oracle queries, there exist adversaries* $\mathcal{B}_{\mathsf{uov}}$ *and* $\mathcal{B}_{\mathsf{mq}}$ *against the NoRS UOV problem with parameters* $(q^\ell, v/\ell, m/\ell, m)$ *and the NoRS QR-MQ problem with parameters* (q, v, m, m, ℓ)*, respectively, such that*

$$\begin{aligned}\mathsf{Adv}(\mathcal{A}_{\mathsf{cma}}) \le{}& \left(2q_{\mathsf{qro}}+1\right)^2\left(\mathsf{Adv}(\mathcal{B}_{\mathsf{uov}})+\mathsf{Adv}(\mathcal{B}_{\mathsf{mq}})\right)\\ &+\frac{3}{2}q'_{\mathsf{sign}}\sqrt{\frac{q'_{\mathsf{sign}}+q_{\mathsf{qro}}+1}{2^\lambda}}\\ &+2\left(q_{\mathsf{qro}}+2\right)\sqrt{\frac{q'_{\mathsf{sign}}-q_{\mathsf{sign}}}{2^\lambda}+\frac{q'_{\mathsf{sign}}+m+1}{2^\lambda}},\end{aligned}$$

where Adv *denotes the advantage of adversaries against each problem and* q'_{sign} *is a bound on the total number of random oracle queries in all the signing queries. The running times of* $\mathcal{B}_{\mathsf{uov}}$ *and* $\mathcal{B}_{\mathsf{mq}}$ *are essentially the same as that of* $\mathcal{A}_{\mathsf{cma}}$*.*

Proof. The proof from the non-invertibility to the EUF-CMA security directly follows [8, Lemma 3]. The reduction from the hardness of the NoRS UOV and QR-MQ problems to the non-invertibility is proven by simply replacing the two problems by the NoRS variants of them in [8, Lemma 5]. □

We note that the underlying problems in Theorem 1 differ from those of the original QR-UOV, as they explicitly incorporate the NoRS structure induced by the biased coefficient distribution. Providing a theoretical proof of the hardness of these NoRS variants is left as future work. Their heuristic hardness, including the effect of the induced bias on known attacks, is discussed in the next subsection. Note that, since each corresponding coefficient of $\mathcal{P}$ and $\mathcal{P}'$ in the NoRS UOV problem is zero with the same probability, the number of zero coefficients does not directly provide a distinguishing advantage between the two maps.

4.2 Concrete security evaluation

In this study, we adopt the same main parameter sets as in the original QR-UOV for the proposed NoRS QR-UOV, corresponding to security levels I, III, and V. This subsection suggests that these parameters also achieve the claimed security levels for NoRS QR-UOV.

In the specification [8], the parameters are chosen such that the complexity of all known attacks on QR-UOV exceeds the criteria for the claimed security levels, as shown in Table 1. In the following, we evaluate the security of NoRS QR-UOV with these parameters by comprehensively analyzing the complexity of these attacks against NoRS QR-UOV. In particular, we discuss the effect of NoRS on the efficiency of the claw-finding attack, the direct attack on the QR-MQ problem, and two key-recovery attacks on the UOV problem, the reconciliation [9] and rectangular MinRank [10, 11] attacks. We do not discuss the Kipnis-Shamir [12] and intersection [10] attacks, since their complexities are significantly higher than those of the other attacks as shown in Table 1, and thus they do not affect the security of the parameters even when NoRS is applied.

Claw-finding attack For a given message representative μ, the claw-finding attack computes $\mathcal{P}(\mathbf{x})$ for multiple $\mathbf{x} \in \mathbb{F}_q^n$ and $\mathsf{Hash}(\mu, r)$ for multiple bit strings r. The attack then forges a signature by finding a collision between these two sets. The complexity of the attack mainly depends on the probability that $\mathcal{P}(\mathbf{x}) = \mathsf{Hash}(\mu, r)$ holds for randomly chosen $\mathbf{x}$ and r. In NoRS QR-UOV, the distribution of $\mathcal{P}$ differs from that in the original scheme, whereas $\mathsf{Hash}(\mu, r)$ outputs a uniformly random element of $\mathbb{F}_q^m$. Therefore, the probability that $\mathcal{P}(\mathbf{x}) = \mathsf{Hash}(\mu, r)$ holds remains the same as in the original QR-UOV, namely q^{-m}.

We also note that NoRS changes the image distribution of $\mathcal{P}(\mathbf{x})$, since the public coefficients are generated from the biased NoRS distribution. However, to the best of our knowledge, no method is

Table 1: The complexity of the existing attacks, the claw finding (Claw), direct (Dir.), reconciliation (Rec.), rectangular MinRank (RM), Kipnis-Shamir (KS), and intersection (Int.) attacks, on the main parameters of QR-UOV in Table 14 of [8]

SL	(q, v, m, ℓ)	complexity					
		Claw	Dir.	Rec.	RM	KS	Int.
I	(127,156,54,3)	201	150	164	158	718	460
III	(127,228,78,3)	286	211	231	219	1056	653
V	(127,306,105,3)	380	279	291	277	1414	851

currently known that improves claw-finding by exploiting the non-uniformity of the image distribution of $\mathcal{P}(\mathbf{x})$. In the standard claw-finding strategy, the success probability is determined by the number of distinct values obtained from evaluations of $\mathcal{P}$, rather than by the precise shape of the image distribution itself. For a fixed set $S_{\mathcal{P}}$ of distinct values obtained as $\mathcal{P}(\mathbf{x})$, the probability that one of M independent hash outputs falls in $S_{\mathcal{P}}$ is

$$1 - \left(1 - \frac{|S_{\mathcal{P}}|}{q^m}\right)^M.$$

Thus, if the bias increases collisions among the values $\mathcal{P}(\mathbf{x})$, it reduces the number of distinct $\mathcal{P}$-side values and more evaluations of $\mathcal{P}$ are needed to obtain the same success probability. Equivalently, a possible decrease in the min-entropy of the output distribution of $\mathcal{P}(x)$ would appear as fewer distinct image values and more collisions among the computed values of $\mathcal{P}(x)$.

Consequently, our analysis suggests that NoRS does not improve the efficiency of the standard claw-finding attack compared with the original QR-UOV. Whether collisions caused by the biased distribution can be exploited to improve the efficiency of any attack is left as future work.

Direct attack We evaluate the complexity of the direct attack on the NoRS QR-MQ problem, which simply solves the MQ system algebraically. In general, the complexity of solving an MQ system with n variables and m equations is given by

$$3\binom{n+2}{2}\binom{n+D}{D}^2,$$

where D denotes the solving degree determined for the given system. Let I be the ideal generated by the given quadratic polynomials, and let $I_{\leq d}$ denote the subspace of I consisting of polynomials of degree at most d. Then, the solving degree D is defined as the minimum degree d such that the codimension of $I_{\leq d}$ in $\mathbb{F}_q[\mathbf{x}]_{\leq d}$ is equal to 1, assuming that the solution is unique.

To compare the complexity of the direct attack against the original and NoRS QR-MQ problems, we compare the solving degrees of the two systems experimentally. In the direct attack, one typically fixes some variables so that the resulting system becomes overdetermined, and then solves the reduced system algebraically. In this setting, the way the variables are fixed may affect how the NoRS-induced bias is reflected in the reduced system. In the NoRS QR-MQ distribution induced from the public-key generation, the coefficients in the $P_{i,1}$ and $P_{i,2}$ parts are directly sampled from D_{NoRS}. In contrast, the coefficients in the $P_{i,3}$ part are obtained through additions and multiplications of independently sampled values, and their zero probability is expected to be closer to that of the uniform distribution. Therefore, fixing variables so that the $P_{i,1}$ part remains in the reduced system is expected to make the NoRS structure be reflected more directly.

Accordingly, in this solving-degree experiment, we use the following reduced NoRS QR-MQ system with n variables and m equations: For each $i \in [m]$, we first sample a symmetric matrix $\widehat{P}_i \leftarrow D_{\mathrm{NoRS}}(q^\ell)^{n/\ell \times n/\ell}$. We then construct $\bar{P}_i \in \mathbb{F}_q^{n\times n}$ by expanding each element $g \in \mathbb{F}_{q^\ell}$ of $\widehat{P}_i$ to an $\ell \times \ell$

Table 2: The comparison of the codimension of $I_{\leq d}$ in $\mathbb{F}_q[\mathbf{x}]_{\leq d}$ for the direct attack against the reduced original QR-MQ problem and the reduced NoRS QR-MQ problem with $q = 127$ and $\ell = 3$

(n, m)	QR-MQ	d						
		2	3	4	5	6	7	8
(12,14)	original	77	273	637	1001	1001	429	1
	NoRS	77	273	637	1001	1001	429	1
(15,18)	original	118	528	1581	3264	4488	3264	1
	NoRS	118	528	1581	3264	4488	3264	1
(15,19)	original	117	512	1463	2736	2907	1	
	NoRS	117	512	1463	2736	2907	1	
(15,20)	original	116	496	1346	2224	1444	1	
	NoRS	116	496	1346	2224	1444	1	

block $W\Phi_g^f$ over $\mathbb{F}_q$. The resulting MQ map is $\mathcal{P}(\mathbf{x}) = \left(\mathbf{x}^\top \bar{P}_1 \mathbf{x}, \ldots, \mathbf{x}^\top \bar{P}_m \mathbf{x}\right)^\top$, and the problem is to solve $\mathcal{P}(\mathbf{x}) = \mathbf{t}$ for a given $\mathbf{t} \in \mathbb{F}_q^m$.

Through our experiments, Table 2 shows that the codimension of $I_{\leq d}$ for $2 \leq d \leq D$ in the reduced NoRS QR-MQ problem coincides with that of the corresponding original QR-MQ problem for some parameters. For each parameter set, we generated 10 independent instances for both the original and NoRS distributions and performed the computations using Magma. For all cases, the ranks of the corresponding Macaulay matrices were identical across all 10 instances. These experiments are performed on small parameters due to time constraints and therefore do not provide direct evidence for the full-size parameter sets. Nevertheless, they are intended to test whether the NoRS-induced bias changes the codimensions of the ideals generated by the corresponding systems, since these codimensions determine the solving degree of the direct attack.

We also note that the coefficient bias slightly affects the number of nonzero monomials in the NoRS QR-MQ problem. In the complexity estimate of the direct attack, each equation contains at most $\binom{n+2}{2}$ nonzero quadratic monomials. Even in the original QR-MQ problem, since each coefficient is sampled uniformly from $\mathbb{F}_q$, the probability that a coefficient is nonzero is $(q-1)/q$. For $q = 127$, this factor is $126/127$, and it is usually ignored in the complexity estimate. In the NoRS case, the zero coefficient occurs with probability $2/128$, so the probability that a coefficient is nonzero becomes $126/128$. Thus, the constant-factor reduction in the expected number of nonzero coefficients is slightly larger than in the uniform case. However, the factor $126/128$ is still very close to 1 and does not affect the claimed security level.

From these analyses, we assume that the direct attack is not substantially improved against the NoRS QR-MQ problem compared with the original QR-MQ problem.

Reconciliation attack The reconciliation attack on the underlying UOV problem with parameters $(q^\ell, v/\ell, m/\ell, m)$ solves the system of

$$\mathbf{x}^\top \bar{P}_i \mathbf{x} = 0 \quad (i \in [m]),$$

where $\bar{P}_i \in \mathbb{F}_{q^\ell}^{n/\ell \times n/\ell}$ denotes the public key matrix of UOV over the extension field. The dimension of the solution space is m/ℓ. Hence, the system can be regarded as one with $v/\ell + 1$ variables and m homogeneous quadratic equations over $\mathbb{F}_{q^\ell}$.

The complexity of the reconciliation attack depends on the solving degree D as in the direct attack. We therefore compare the solving degree of the attack for the original and NoRS UOV problems by experimentally evaluating the codimension of the ideal generated by the equations, as shown in Table 3. Unlike in the direct attack case, we consider the subspace $I_d \subseteq I$ consisting of degree-d polynomials,

Table 3: The comparison of the codimension of I_d in $\mathbb{F}_{q^\ell}[\mathbf{x}]_d$ for the reconciliation attack against the original and NoRS UOV problems with parameters $(q^\ell, v/\ell, m/\ell, m)$ where $q = 127$ and $\ell = 3$

(v,m)	UOV	d					
		2	3	4	5	6	7
(30,12)	original	54	154	275	297	132	1
	NoRS	54	154	275	297	132	1
(33,15)	original	63	184	300	168	1	
	NoRS	63	184	300	168	1	
(36,18)	original	73	221	335	1		
	NoRS	73	221	335	1		
(36,21)	original	70	182	119	1		
	NoRS	70	182	119	1		

since the system to be solved in the key recovery attacks is homogeneous. Table 3 shows that the solving degree takes the same value for the reconciliation attacks on the original and NoRS UOV problems with parameters $(q^\ell, v/\ell, m/\ell, m)$ for the tested parameter values. For each parameter set, we generated 10 independent instances for both cases and performed the computations using Magma. For all cases, the ranks of the corresponding Macaulay matrices were identical across all 10 instances. These experiments are performed on small parameters due to time constraints and therefore do not provide direct evidence for the full-size parameter sets. From the results, we assume that the reconciliation attack is not improved against the NoRS UOV problem in larger parameter ranges, compared with the original UOV problem.

As in the previous analyses, the NoRS-induced coefficient bias only changes the number of nonzero coefficients by a small constant factor; for coefficients over $\mathbb{F}_{q^\ell}$, this factor is even closer to 1 than the factor $126/128$ in the direct-attack analysis, and hence it does not change the claimed security level.

Rectangular MinRank attack We analyze the complexity of the rectangular MinRank attack on the underlying NoRS UOV problem with parameters $(q^\ell, v/\ell, m/\ell, m)$. The attack consists of the following two steps:

1. Transform the public key matrices over $\mathbb{F}_{q^\ell}$ into a set of n/ℓ matrices in $\mathbb{F}_{q^\ell}^{n/\ell \times m}$.
2. Find a linear combination of the n/ℓ matrices having rank v/ℓ by solving a MinRank problem.

From the viewpoint of complexity, the second step is dominant, and the support minors method [13] is known to be one of the most efficient approaches to solving the MinRank problem. The method introduces $v/\ell + 1$ linear variables $\mathbf{x}$ corresponding to elements of the secret key S and $\binom{m}{v/\ell}$ support variables $\mathbf{c}$. For these variables, we solve a system consisting of $\binom{m}{v/\ell+1}n/\ell$ homogeneous bilinear equations in $\mathbf{x}$ and $\mathbf{c}$, and m homogeneous quadratic equations in $\mathbf{x}$. The complexity of the support minors approach depends on the solving degree of the resulting system, as in the direct and reconciliation attacks. In practice, the system is typically solved by adjusting only the degree of $\mathbf{x}$ while fixing the degree of $\mathbf{c}$ to 1 for efficiency.

We then estimate the complexity of the rectangular MinRank attack on the NoRS UOV problem by comparing the solving degree D. Table 4 shows the codimension of $I_{d,1}$ in $\mathbb{F}_{q^\ell}[\mathbf{x}, \mathbf{c}]_{d,1}$, where $I_{d,1}$ denotes the set of polynomials in the ideal I generated by the equations to be solved whose degree in $\mathbf{x}$ is d and whose degree in $\mathbf{c}$ is 1. The table shows that the solving degree takes the same value for the rectangular MinRank attacks on the original and NoRS UOV problems with parameters $(q^\ell, v/\ell, m/\ell, m)$ for the tested parameter values. For each parameter set, we generated 10 independent instances for both cases and performed the computations using Magma. For all cases, the ranks of the corresponding Macaulay matrices were identical across all 10 instances. These experiments are performed on small parameters due to time constraints and therefore do not provide direct evidence for the full-size parameter

Table 4: The comparison of the codimension of $I_{d,1}$ in $\mathbb{F}_{q^\ell}[\mathbf{x}, \mathbf{c}]_{d,1}$ for the rectangular MinRank attack against the original and NoRS UOV problems with parameters $(q^\ell, v/\ell, m/\ell, m)$ where $q = 127$ and $\ell = 3$

(v, m)	UOV	d			
		2	3	4	5
(15,6)	original	49	21	1	
	NoRS	49	21	1	
(18,9)	original	1			
	NoRS	1			
(21,9)	original	316	1		
	NoRS	316	1		
(24,9)	original	226	369	246	1
	NoRS	226	369	246	1

sets. From the results, we assume that the rectangular MinRank attack is not improved against the NoRS UOV problem in larger parameter ranges, compared with the original UOV problem.

Note that the NoRS-induced coefficient bias only changes the number of nonzero coefficients by a small constant factor; for coefficients over $\mathbb{F}_{q^\ell}$, this factor is even closer to 1 than the factor $126/128$, and hence it does not change the claimed security level.

We also experimentally checked whether the rank of the target matrix in NoRS QR-UOV decreases due to the coefficient bias. For each of the level I, III, and V parameter sets, we generated 100 independent instances and computed the rank of the corresponding target matrix. In all trials, the rank was equal to v/ℓ, and no rank decrease was observed.

We studied the impact of the coefficient bias introduced by NoRS on the security of QR-UOV with respect to existing attacks. These analyses do not constitute a proof that the NoRS assumptions have the same hardness as the original QR-UOV assumptions. Rather, they show that we did not observe any reduction in the complexity of the considered attacks for the proposed parameter sets.

5 Efficient implementation of NoRS QR-UOV verification

As described in Section 2.2, QR-UOV verification consists of public-key expansion followed by dot-product evaluation. In Algorithm 1, for each coordinate i, the verifier regenerates $(\boldsymbol{p}_{i,1}, \boldsymbol{p}_{i,2})$ from $\mathsf{seed}_{\mathsf{pk}}$ and then evaluates $t'_i = \boldsymbol{u}_1 \cdot \boldsymbol{p}_{i,1} + \boldsymbol{u}_2 \cdot \boldsymbol{p}_{i,2} + \boldsymbol{u}_3 \cdot \boldsymbol{p}_{i,3}$. Therefore, verification speed largely depends on how efficiently the expanded coefficients are generated and consumed inside this loop. NoRS is most effective in this part of the verification procedure, because it simplifies public-key expansion, enabling a fusion of coefficient generation and accumulation.

We focus on the core computation $\boldsymbol{u} \cdot \boldsymbol{p}^{(i)}$, where $\boldsymbol{p}^{(i)}$ is a coefficient vector generated from $(\mathsf{seed}_{\mathsf{pk}}, i)$. In verification, this computation is used for $\boldsymbol{p}_{i,1}$ and $\boldsymbol{p}_{i,2}$, while the contribution of $\boldsymbol{p}_{i,3}$ is handled separately because it is already stored in the compressed public key. Algorithm 2 computes this quantity without materializing $\boldsymbol{p}^{(i)}$ in memory. Instead, it accumulates the dot product while the coefficients are being generated.

The kernel works block by block. For each block index j, it generates one AES output block from $(\mathsf{seed}_{\mathsf{pk}}, i \parallel j)$, converts it into 16 coefficients, multiplies them by the corresponding 16-entry segment of the precomputed vector $\boldsymbol{u}$ derived from signature, and immediately adds the result to the running sum. In this way, generation and consumption are fused into a single loop. Since $\boldsymbol{u}$ is padded in advance to a multiple of 16, the kernel needs no separate remainder path. This is possible because of removal of rejection sampling. In standard QR-UOV, coefficients derived from the AES output cannot be used

Algorithm 2 Fused expansion-and-dot-product kernel for NoRS QR-UOV verification

Input: coefficient vector $\boldsymbol{u} \in \mathbb{F}_q^{t_{\mathrm{pad}}}$, public seed $\mathsf{seed}_{\mathsf{pk}}$, index $i \in [m]$
Output: $\boldsymbol{u} \cdot \boldsymbol{p}^{(i)} \in \mathbb{F}_q$, where $\boldsymbol{p}^{(i)}$ is the coefficient vector derived from $(\mathsf{seed}_{\mathsf{pk}}, i)$

1: $acc \leftarrow 0$
2: **for** $j \leftarrow 0$ **to** $t_{\mathrm{pad}}/16 - 1$ **do**
3: $\quad \boldsymbol{r} \leftarrow \mathsf{AES}(\mathsf{seed}_{\mathsf{pk}}, i \parallel j)$ ▷ $\boldsymbol{r} \in \{0, \ldots, 255\}^{16}$
4: $\quad \boldsymbol{p} \leftarrow \boldsymbol{r}\ \&\ q$ ▷ $\boldsymbol{p} \in \{0, \ldots, q\}^{16}$
5: $\quad acc \leftarrow acc + \boldsymbol{u}[16j : 16j + 15] \cdot \boldsymbol{p}$
6: **end for**
7: **return** $acc \bmod q$

Note. Since $q = 2^k - 1$, $\boldsymbol{p} \leftarrow \boldsymbol{r}\ \&\ q$ denotes byte-wise extraction of the lower k bits. Entries equal to q are kept as non-canonical representatives of $0 \in \mathbb{F}_q$.

immediately, because each occurrence of the out-of-range value q in the required prefix must be replaced by a later valid value. As discussed in Section 2.3, this replacement step introduces extra scanning and irregular control flow. In NoRS, by contrast, the exceptional value is mapped to 0 and can be kept as a non-canonical representative of zero in the implementation. Since $q = 2^k - 1$ with $k \in \{3, 5, 7\}$, each byte yields one coefficient candidate after applying a fixed byte-wise mask, and hence each AES output block yields 16 candidates. This regular structure allows expansion and accumulation to be merged into a single kernel.

The structure is well-suited to SIMD execution on modern x86 processors. Our implementation uses AES-NI for block generation and AVX2 for vectorized dot-product computation. In the vectorized dot-product routine, products of 8-bit operands are accumulated into 32-bit lanes. The byte value q is treated as a non-canonical representative of $0 \in \mathbb{F}_q$. Since $q \bmod q = 0$, any product involving this value contributes zero after the final reduction modulo q. Since each integer product is at most $q(q-1)$, the accumulator is at most $t_{\mathrm{pad}}q(q-1)$. For all QR-UOV parameter sets considered in this paper, this upper bound fits in a 32-bit signed integer. Therefore, no modular reduction is required inside the accumulation loop; it is sufficient to reduce the final accumulator modulo q after the loop. Except for loading the precomputed vector $\boldsymbol{u}$, the main part of the kernel stays in registers, so the generated coefficients are consumed without being written back to memory. This removes the temporary buffer needed in the rejection-sampling-based implementation and reduces memory accesses during verification. A remaining practical bottleneck is the latency of AES-NI instructions. To mitigate this cost, we unroll the loop and interleave multiple independent AES blocks, increasing instruction-level parallelism and hiding part of the AES latency.

6 Performance analysis

6.1 Experimental setup

We benchmarked implementations on an Intel Skylake platform with AVX2 and AES-NI support. The processor was a Xeon Platinum 8124M on an Amazon EC2 c5n.metal instance. The benchmark thread was pinned to a fixed logical CPU. CPU frequency scaling was configured so that the pinned logical CPU ran at a fixed frequency of 3.0 GHz, and its SMT sibling logical CPU was offlined to reduce measurement noise. Cycle counts were measured with the libcpucycles [14] library using hardware performance counters. Each implementation was executed 1,000 times, and we report the median cycle count in Mcycles. The benchmark programs were compiled with GCC 13.3 on Ubuntu 24.04 LTS, except that the QR-UOV matrix multiplication routines were built with Clang 18.1. All builds used the `-O3` optimization flag.

Our implementation was derived from the AVX2-optimized QR-UOV implementation in the previous paper [4]. This work focused on performance optimization and did not aim to provide constant-time

Table 5: Benchmark results for QR-UOV, NoRS QR-UOV, UOV, and MAYO using Round-2 parameter sets and standard PRG settings (Mcycles)

SL	Scheme	KeyGen	Sign	Verify
I	QR-UOV	2.85	0.83	0.43
	NoRS QR-UOV	2.73	0.72	0.30
	UOV Ip-pkc+skc	3.02	1.99	0.27
	MAYO1	0.24	0.70	0.29
III	QR-UOV	10.23	2.45	1.37
	NoRS QR-UOV	9.88	2.11	0.96
	UOV III-pkc+skc	16.92	10.97	1.14
	MAYO3	0.56	1.45	0.65
V	QR-UOV	28.65	6.23	3.62
	NoRS QR-UOV	27.82	5.46	2.62
	UOV V-pkc+skc	54.95	24.93	2.51
	MAYO5	1.36	3.51	1.50

guarantees. The source code of our implementation will be made publicly available upon publication.

We compared our implementation with the official implementations of UOV[2] [5] and MAYO[3] [7]. For UOV, we selected the pkc+skc variant. We use these schemes as representative compressed-key multivariate reference schemes. For UOV and MAYO, we used the Round-2 parameter sets and the original AVX2 build configurations. The only modification was to add the benchmarking harness needed to call the target routines and measure their cycle counts under the same methodology.

In the main benchmarks, we used the standard Round-2 settings of each scheme. In these settings, UOV and MAYO use AES-128 as the PRG for public-key expansion, whereas QR-UOV uses AES-128, AES-192, and AES-256 at security levels I, III, and V, respectively. To separate the effect of the PRG choice from the cost of signature verification, we also report supplementary verification benchmarks in which the PRG is fixed to AES-128 for all security levels. We restrict these supplementary benchmarks to verification, because their purpose is to isolate the effect of the PRG choice on the verification cost, where public-key expansion is the dominant component. We further include results with four-round AES-128 (AES4), which is optional in the UOV specification and non-standard for QR-UOV and MAYO. The AES4 results further reduce the relative cost of the AES-based PRG and help indicate how much of the verification cost remains outside pseudorandom generation.

6.2 Benchmark results

Table 5 reports the main benchmark results under the standard Round-2 settings of each scheme. The proposed NoRS variant consistently improves the performance of QR-UOV at all security levels. The largest gain appears in signature verification, where the speed-up is about 1.4× across all security levels.

The improvement in signing is smaller but still consistent, at about 1.15× across all security levels. In contrast, the effect on key generation is limited, with only a small reduction at each security level. The table suggests that NoRS QR-UOV has verification performance comparable to UOV.

Table 6 gives supplementary verification results under the AES-128-only and AES4 settings. In verification, the cost of NoRS QR-UOV becomes close to those of UOV and MAYO at security level I. At security levels III and V, it is lower than that of UOV and higher than that of MAYO in this benchmark setting. AES4 settings show the same qualitative trend more clearly: NoRS consistently improves QR-UOV at all security levels, and the largest gain appears in verification.

[2] `https://github.com/pqov/pqov` commit `49dc53c0`

[3] `https://github.com/PQCMayo/MAYO-C` commit `4b7cd94c`

Table 6: Supplementary signature-verification benchmarks under AES-128-only and AES4 settings (Mcycles)

SL	Scheme	AES-128 Verify	AES4 Verify
I	QR-UOV	0.43	0.28
	NoRS QR-UOV	0.30	0.15
	UOV Ip-pkc+skc	0.27	0.15
	MAYO1	0.29	0.24
III	QR-UOV	1.22	0.79
	NoRS QR-UOV	0.87	0.42
	UOV III-pkc+skc	1.14	0.61
	MAYO3	0.65	0.51
V	QR-UOV	2.91	1.84
	NoRS QR-UOV	2.08	0.96
	UOV V-pkc+skc	2.51	1.29
	MAYO5	1.50	1.18

The AES-128 setting fixes the PRG to AES-128 at all security levels; for QR-UOV, this differs from the Round-2 standard setting at security levels III and V. AES4 denotes AES-128 reduced to four rounds; it is a supplementary reduced-round setting, optional in the UOV specification and non-standard for QR-UOV and MAYO.

6.3 Profiling of verification routine

To explain the observed verification speedup, we profiled the verification routine at security level I by separating its cost into AES-based pseudorandom generation, the kernel, and the remaining operations. Table 7 reports the results under the same environment as in Section 6.1. Here, NoRS refers to the fused implementation described in Section 5, while the materialized variant is included only to separate the effect of fusion from that of removing rejection sampling. The total times in Table 7 differ slightly from the corresponding Verify times in Table 5 and Table 6, because they were measured separately with a slightly different profiling methodology.

The dominant cost lies in the AES-based pseudorandom generation and kernel parts, whereas the remaining operations stay almost constant across all variants. In the AES-128 baseline, these two parts account for about 91% of the total verification time. This confirms that QR-UOV verification is dominated by public-key expansion and the closely related kernel computation.

The main gain of NoRS already appears before fusion. Moving from the baseline to NoRS (materialized) reduces the total verification time by about 24% in the AES-128 setting and 35% in the AES4 setting, while leaving the AES cost almost unchanged. At the same time, the Kernel cost drops by about 73% in both settings, showing that the primary effect of NoRS is to eliminate the rejection-sampling overhead. The fused kernel provides a further reduction of about 9% in the AES-128 setting and 14% in the AES4 setting by avoiding coefficient materialization. Overall, removing rejection sampling provides the larger gain, and fusion gives an additional improvement.

7 Discussion and conclusion

Overall, the results of this paper show that NoRS is a practical trade-off for QR-UOV. By confining the removal of rejection sampling to public-key expansion, NoRS introduces a slight bias in the generated coefficients, but for the proposed parameter sets, the current analysis did not find a reduction in the

Table 7: Profiling breakdown of QR-UOV verification at security level I (Kcycles)

Variant	AES	Kernel	Others	Total
Baseline AES-128	248.9	127.1	37.8	413.7
NoRS AES-128 (materialized)	241.9	34.1	37.4	313.4
NoRS AES-128	252.2		34.1	286.3
Baseline AES4	103.6	125.4	38.0	266.9
NoRS AES4 (materialized)	101.9	33.7	37.9	173.4
NoRS AES4	115.0		34.0	148.9

The Kernel column includes rejection sampling and dot-product accumulation in the baseline, but only dot-product accumulation in the materialized NoRS variants. For the NoRS variants without materialization, expansion and dot-product accumulation are fused and thus reported jointly.

Table 8: Total bytes of public-key coefficients obtained by expansion

Scheme	SL = I	SL = III	SL = V
QR-UOV $(127, 3)$	374,868	1,147,068	2,779,245
UOV	234,872	1,036,224	2,422,464
MAYO	144,495	376,164	841,847

For UOV at security level I, we use the uov-Ip parameter set.

considered attacks. At the same time, the implementation becomes consistently faster, with the largest improvement in signature verification.

From the security viewpoint, the main concern is the coefficient bias introduced by deterministically mapping the unique rejected value to 0. Our analyses suggest that, for the proposed parameter sets, this bias does not meaningfully change the difficulty of the considered attacks, either asymptotically or concretely. In particular, the claw-finding, direct, reconciliation, and rectangular MinRank attacks exhibit essentially the same behavior as in the original QR-UOV. These results suggest that the induced bias is practically acceptable for the proposed parameter sets.

From the implementation viewpoint, the practical significance of NoRS lies mainly in verification. Since $\mathsf{Expand}_{\mathsf{pk}}$ is invoked in key generation, signing, and verification in the QR-UOV specification, all three routines benefit from NoRS. The benchmark results show that the benefit is consistent across all security levels and is much larger for verification than for key generation or signing.

The large speedup in verification can be explained by the structure of QR-UOV. Since verification is dominated by public-key expansion, simplifying coefficient generation has a strong practical impact. This also clarifies the comparison with UOV and MAYO. Unlike these schemes, which use characteristic-two fields, QR-UOV uses odd base-field sizes and therefore incurs rejection sampling during public-key expansion. Thus, the observed improvement reflects the reduction of an expansion overhead that is specific to QR-UOV among the schemes compared here. Moreover, Table 8 shows that QR-UOV generates more data during public-key expansion than UOV and substantially more than MAYO at all security levels, which helps explain why the benefit of NoRS is large in practice.

The supplementary AES4 results provide a complementary perspective. Once the cost of public-key expansion is reduced, QR-UOV becomes much more competitive with both UOV and MAYO in verification. This suggests that much of its practical overhead comes from coefficient generation during expansion rather than from the verification core itself.

Although our benchmarks are specific to x86 processors with AES-NI and AVX2, the NoRS idea itself

is architecture-independent, since it simplifies coefficient generation at the algorithmic level. Architectures with efficient AES support and vectorized arithmetic may also benefit from the same simplification, but quantifying that benefit is left to future work.

More broadly, this work suggests that carefully scoped removal of rejection sampling can be a useful design option in cryptographic schemes over odd-prime fields, provided that both their security impact and their implementation benefit are evaluated carefully.

References

[1] H. Furue, Y. Ikematsu, Y. Kiyomura, and T. Takagi, "A new variant of unbalanced Oil and Vinegar using quotient ring: QR-UOV," ASIACRYPT 2021, Part IV, ed. M. Tibouchi and H. Wang, LNCS, vol.13093, pp.187–217, Springer, Cham, Dec. 2021.

[2] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced Oil and Vinegar signature schemes," EUROCRYPT'99, ed. J. Stern, LNCS, vol.1592, pp.206–222, Springer, Berlin, Heidelberg, May 1999.

[3] G. Alagic, M. Bros, P. Ciadoux, D. Cooper, Q. Dang, T. Dang, J.M. Kelsey, J. Lichtinger, C.A. Miller, D. Moody, R. Peralta, R. Perlner, A. Robinson, H. Silberg, D. Smith-Tone, N. Waller, and Y.K. Liu, "Status report on the first round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process," tech. rep., National Institute of Standards and Technology, 2024.

[4] H. Amagasa, R. Ueno, and N. Homma, "AVX2 implementation of QR-UOV for modern x86 processors." Cryptology ePrint Archive, Report 2025/1599, 2025.

[5] W. Beullens, M. Chen, J. Ding, B. Gong, M.J. Kannwischer, J. Patarin, B. Peng, D. Schmidt, C. Shih, C. Tao, and B. Yang, "UOV — Unbalanced Oil and Vinegar," tech. rep., National Institute of Standards and Technology, 2024.

[6] L. Wang, C. Chou, J. Ding, Y. Kuan, J.A. Leegwater, M. Li, B. Tseng, P. Tseng, and C. Wang, "SNOVA," tech. rep., National Institute of Standards and Technology, 2024.

[7] W. Beullens, F. Campos, S. Celi, B. Hess, and M.J. Kannwischer, "MAYO," tech. rep., National Institute of Standards and Technology, 2024.

[8] R. Akiyama, H. Furue, Y. Ikematsu, F. Hoshino, K. Kinjo, H. Kosuge, S. Nakamura, S. Orihara, T. Takagi, and K. Yamakoshi, "QR-UOV," tech. rep., National Institute of Standards and Technology, 2024.

[9] J. Ding, B. Yang, C.O. Chen, M. Chen, and C. Cheng, "New differential-algebraic attacks and reparametrization of Rainbow," ACNS 2008, LNCS, vol.5037, pp.242–257, Springer, 2008.

[10] W. Beullens, "Improved cryptanalysis of UOV and Rainbow," EUROCRYPT 2021, LNCS, vol.12696, pp.348–373, Springer, 2021.

[11] H. Furue and Y. Ikematsu, "A new security analysis against MAYO and QR-UOV using rectangular MinRank attack," IWSEC 2023, LNCS, vol.14128, pp.101–116, Springer, 2023.

[12] A. Kipnis and A. Shamir, "Cryptanalysis of the oil and vinegar signature scheme," CRYPTO 1998, LNCS, vol.1462, pp.257–266, Springer, 1998.

[13] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R.A. Perlner, D. Smith-Tone, J.P. Tillich, and J.A. Verbel, "Improvements of algebraic attacks for solving the rank decoding and MinRank problems," ASIACRYPT 2020, LNCS, vol.12491, pp.507–536, Springer, 2020.

[14] D.J. Bernstein, "libcpucycles." `https://cpucycles.cr.yp.to/`, 2024.