Practical and Theoretical Cryptanalysis of VOX

Hao Guo[1,2], Yi Jin[3], Yuansheng Pan[3],
Xiaoou He[3], Boru Gong[3] and Jintai Ding[1,2]

[1] Beijing Institute of Mathematical Sciences and Applications, Beijing, China
[2] Yau Mathematical Sciences Center, Tsinghua University, Beijing, China
[3] CCBFT, Shanghai, China

Abstract. VOX is a UOV-like hash-and-sign signature scheme from the Multivariate Quadratic (MQ) family, which has been submitted to NIST Post-Quantum Cryptography Project, in response to NIST's Call for Additional Digital Signature Schemes for the PQC Standardization Process. In 2023, the submitters of VOX updated the sets of recommended parameters of VOX, due to the rectangular MinRank attack proposed by Furue and Ikematsu.
In this work we demonstrate the insecurity of the updated VOX, from both the practical and the theoretical aspects, and more works need be done with respect of the security analysis of VOX.
First, we conduct a practical MinRank attack against VOX, which uses multiple matrices from matrix deformation of public key to form a large rectangular matrix and evaluate the rank of this new matrix. By using Kipnis–Shamir method and Gröbner basis calculation only instead of support-minors method, our experiment shows it could recover, *within two seconds*, the secret key of almost every updated recommended instance of VOX. And the analysis about the rationale behind the power of this practical attack is still on its way.
Moreover, we propose a theoretical analysis on VOX by expressing public/secret key as matrices over a smaller field to find a low-rank matrix, resulting in a more precise estimation on the concrete hardness of VOX; for instance, the newly recommended VOX instance claimed to achieve NIST security level 3 turns out to be 69-bit-hard, as our analysis shows.

Keywords: PQC, MPKC, VOX

1 Introduction

The UOV signature scheme has been introduced for more than 20 years.However, UOV and its variants suffer from long public key length. Therefore the researchers has been devoted to compressing the public key size of UOV as well as its variants. Recently NIST announced an additional round for post-quantum signatures and received about 40 submissions. Among the submissions seven of them are UOV-like schemes: MAYO [BCC+23], PROV [GCF+23], QR-UOV [FIH+23], SNOVA [WCD+23], TUOV [DGG+23], UOV [BCD+23] and VOX [PCF+23].

In this work we concentrate on the VOX scheme [PCF+23], which was proposed by Patarin et al., and it combines the idea of QR-UOV [FIKT21] and plus modification [FmRPP22]. After the publication of VOX, Furue and Ikematsu [FI23] proposed an equivalent key recovery attack using rectangular MinRank attack. Rectangular MinRank attack was proposed by Beullens in [Beu21a], and the idea has also been found in [TPD21]. In [FI23] the authors showed that MAYO and QR-UOV remains secure under the rectangular MinRank attack. However VOX turned out to be vulnerable under their traditional rectangular MinRank attack, due to the fact that $O > t$ in its previously recommended

Table 1: Experiment result of our practical attack.

λ	q	O	V	c	t	Running time (second)	Total Memory Usage (MB)
128	251	4	5	13	6	0.170	32.09
		5	6	11	6	0.510	32.09
		6	7	9	6	27357.799	6147.06
192	1021	5	6	15	7	0.440	32.09
		6	7	13	7	0.790	32.09
		7	8	11	7	26.170	157.69
256	4093	6	7	17	8	1.240	64.12
		7	8	14	8	1.870	64.12
		8	9	13	8	51.530	256.00

parameters, where O and t denote the number of oil variables and that of random polynomials, respectively. Consequently, the submitters of VOX updated their recommended parameters by requiring that $O \leq t$ and claimed that the new design will withstand the rectangular MinRank attack [MPC+23].

In this work we demonstrate that VOX equipped with its newly updated recommended parameters [MPC+23] is still *insecure* from both the practical and theoretical aspects, and more work should be done in respect of security analysis of VOX.

Practical attack against VOX. First, the main contribution of this work is a MinRank attack against VOX even if $t \geq O$. Its general idea is to concatenate l matrices from matrix deformation [INT23] of public key matrices vertically and evaluate the rank of this new matrix. By observing that columns of central map shuffle consistently thanks to matrix deformation formula, we find that such vertical contatenation of multiple matrices can not only be done on rectangular central map to form a matrix of rank at most $lV + t$, but also be done on rectangular public key matrices and also form a matrix of the same rank.

Compared with the attack in [FI23] which only uses one matrix from matrix deformation, our attack uses multiple matrices to form the target matrix, making our attack work as long as $O \leq t < O(O+1)/2$. Moreover, when solving this MinRank instance, we use Kipnis–Shamir method instead of support-minors method, and we solve the equations generated by Kipnis–Shamir method using only Gröbner basis calculation, which is in sharp contrast with other algorithms for the MinRank problem. The power of our attack can be fully demonstrated by the following experiment: when running on a server with a 2.40GHz CPU and 32GB memory, the first attack can quickly recover, within 1 minute, the secret key of almost every VOX recommended instance; in particular, it takes less than 2 seconds for six out of nine recommended instances of VOX. However, we do not know why our first attack can break VOX in such an efficient manner, and more work need to be done in terms of its theoretical analysis.

Theoretical analysis against VOX. Furthermore, we propose a theoretical analysis against VOX, which could be traced back to the QR-structure in VOX. As shown in Section 4, when the dimension c has a proper factor, say c_1, the field extension $\mathbb{F}_q \subset \mathbb{F}_{q^c}$ in the VOX has a nontrivial intermediate field $\mathbb{F}_{q^{c_1}}$, and the public/secret keys could be seen as matrices over this intermediate field obviously; moreover, direct verification shows that when the degree of extension $[\mathbb{F}_{q^c} : \mathbb{F}_{q^{c_1}}]$ is larger than t/O, we can always construct from the secret key a matrix that is not full-rank, and then use Kipnis-Shamir method to solve this MinRank problem over this intermediate extension field. Compared with previous MinRank attacks, our theoretical analysis aims to find low-rank matrices in an intermediate field by fully utilizing properties of the QR-structure, provided that c

Table 2: Estimated complexity of our theoretical attack.

λ	q	$O = m/c$	$V = v/c$	c	c_1	t	d	D_{mgd}	$\log_2 C$
128	251	6	7	9	3	6	2	12	112.46
	251	5	6	10	5	6	1	6	49.64
192	1021	5	6	15	5	7	1	8	69.48
256	4093	7	8	14	7	8	1	5	48.04

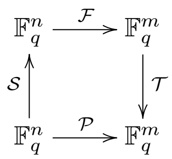
is composite. The strength of our second attack can be gleaned from the fact that for the the newly updated recommended parameter sets of VOX claimed to achieve NIST security 1, 3, and 5, their concrete hardness are actually 112-, 69-, and 48-bits-hard, respectively.

Organization. Our paper is organized as follows. Section 2 contains some preliminaries including the VOX scheme, MinRank problem and rectangular MinRank attack. In Section 3, we first introduce our padded MinRank attack, then show its practical performance against VOX parameters, and finally give our explanation of why it works. In Section 4, we first show the idea of intermediate field attack, explaining its construction, then give our hypothetical complexity analysis for the parameters that this attack can be used on. We conclude this work in Section 5.

2 Preliminaries

2.1 About the VOX scheme

Generally speaking, a UOV-like digital signature scheme makes $\mathcal{P}$ its public key and $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ the private key with $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{S}$ where $\mathcal{S} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ and $\mathcal{T} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ are both invertible linear transformations, and $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ consists of m homogeneous quadratic polynomials $f_1, \ldots, f_m$ that can be somehow efficiently invertible. For simplicity, we would identify maps $\mathcal{S}, \mathcal{T}, \mathcal{F}, \mathcal{P}$ with square matrices $\mathbf{S} \in \mathrm{GL}_n(\mathbb{F}_q), \mathbf{T} \in \mathrm{GL}_m(\mathbb{F}_q)$ and symmetric matrices $\mathbf{F} \in \mathsf{Mat}_n(\mathbb{F}_q), \mathbf{P} \in \mathsf{Mat}_m(\mathbb{F}_q)$ respectively.



To invert $\mathcal{F}$ efficiently, OV polynomial comes to attention. An $(n, n-m)$-OV polynomial f_k can be defined as

$$f_k(x_1, \ldots, x_n) = \sum_{i=1}^{n-m} \sum_{j=i}^{n} a_{ij}^{(k)} x_i x_j$$

with $a_{ij}^{(k)} \in \mathbb{F}_q$. Notice that f_k is linear in $x_{n-m+1}, \ldots, x_n$ when $x_1, \ldots, x_{n-m}$ are fixed. Then we say there are $v = n - m$ vinegar-variables $x_1, \ldots, x_v$ and $o = m$ oil-variables $x_{v+1}, \ldots, x_{v+o}$.

VOX is a UOV-like scheme that constructs the secret key $\mathcal{F}$ by mixing t totally random quadratic polynomials and $o - t$ OV polynomials with quotient ring structure. Let c be a common divisor of o and v, we denote $O = o/c$, $V = v/c$ and $N = n/c$. Then there are V vinegar-variables, O oil-variables and o equations over $\mathbb{F}_{q^c}$ utilizing the QR-structure. Specifically, we have private key $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ with $\mathcal{S} : \mathbb{F}_{q^c}^N \to \mathbb{F}_{q^c}^N$ and $\mathcal{T} : \mathbb{F}_{q^c}^o \to \mathbb{F}_{q^c}^o$ are both invertible linear transformations, and $\mathcal{F} : \mathbb{F}_{q^c}^N \to \mathbb{F}_{q^c}^o$ consists of t totally random quadratic

Table 3: Current parameters of VOX.

λ	q	$O = m/c$	$V = v/c$	c	t
128	251	4	5	13	6
		5	6	11	6
		6	7	9	6
192	1021	5	6	15	7
		6	7	13	7
		7	8	11	7
256	4093	6	7	17	8
		7	8	14	8
		8	9	13	8

polynomials and $o - t$ (N, V)-OV polynomials. Notice that $\mathcal{T}$ has matrix representation $\mathbf{T} \in \mathrm{GL}_o(\mathbb{F}_q)$. And for simplicity, we can denote the public key as $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{S} : \mathbb{F}_{q^c}^N \to \mathbb{F}_{q^c}^o$.

$$\begin{array}{ccccc}
\mathbb{F}_{q^c}^N & \xrightarrow{\mathcal{F}} & \mathbb{F}_{q^c}^o & \xrightarrow{\mathsf{Tr}^{\oplus o}} & \mathbb{F}_q^o \\
\uparrow \mathcal{S} & & \downarrow \mathcal{T} & & \downarrow \mathcal{T} \\
\mathbb{F}_{q^c}^N & \xrightarrow{\mathcal{P}} & \mathbb{F}_{q^c}^o & \xrightarrow{\mathsf{Tr}^{\oplus o}} & \mathbb{F}_q^o
\end{array}$$

Here we list the current parameters given in [MPC$^+$23] in Table 3.

2.2 The MinRank problem

Put it simply, the MinRank problem asks for a linear (or affine) combination of given matrices that has a small rank. This problem is first abstracted by Courtois [Cou01], where he generalized the problem of Syndrome Decoding from coding theory. The problem we are interested is the search version of the MinRank problem:

Definition 1 (Homogeneous MinRank problem)**.** Let $\mathbf{M}_1, \ldots, \mathbf{M}_K$ be some m-by-n matrices over a *finite* field $\mathbb{F}_q$, and let $r < \min(m, n)$. The problem asks for $x_1, \ldots, x_K \in \mathbb{F}_q$ which are not all zero, such that

$$\mathbf{M} := \sum_{k=1}^{K} x_k \mathbf{M}_k$$

has rank no more than r.

We denote the set of problems with parameter (m, n, K, r, q) as $\mathrm{MR}(m, n, K, r, q)$. When the field is clear from context we also omit q. There is also the inhomogeneous version:

Definition 2 (Inhomogeneous MinRank problem)**.** Let $\mathbf{M}_0; \mathbf{M}_1, \ldots, \mathbf{M}_K$ be some m-by-n matrices over a *finite* field $\mathbb{F}_q$, and let $r < \min(m, n)$. The problem asks for $x_1, \ldots, x_K \in \mathbb{F}_q$, such that

$$\mathbf{M} := \mathbf{M}_0 + \sum_{k=1}^{K} x_k \mathbf{M}_k$$

has rank no more than r.

We denote the set of problems with parameter (m, n, K, r, q) as $\overline{\mathrm{MR}}(m, n, K, r, q)$. When the field is clear from context we also omit q.

In homogeneous case, we require that all the $\mathbf{M}_k$'s are of rank at least $r+1$; In inhomogeneous case, we require that $\mathbf{M}_0$ is of rank at least $r+1$. This is to avoid trivial solutions.

2.3 Combinatorial and algebraic methods for solving the MinRank problem

Courtois mentioned in [Cou01] that the MinRank problem is NP-hard via reduction from syndrome decoding problem of a linear error correcting code which is NP-complete. Faugère [FLP08] on another hand gives a reduction from rank decoding problem, also showing its hardness. Nonetheless, there have been many methods to solve the MinRank problem. These methods fall into two categories: combinatorial method, and algebraic method.

Kernel attack [GC00] is the first method proposed to solve the MinRank problem. It is proposed by Goubin and Courtois. The idea is to choose vectors $\mathbf{y}_k \in \mathbb{F}_q^n$ randomly, hoping they could fall into the kernel of $\mathbf{M}$, the linear (affine) combination of given matrices, then solve for the coefficient x_k's using the linear equations $\mathbf{M}\mathbf{y}_k = \mathbf{0}$. This is a combinatorial method, and the complexity of kernel attack is $O(q^{\lceil K/m \rceil r} K^3)$.

Minors attack [FDS10] is the simple algebraic method, which takes out all $(r+1)$-minors of $\mathbf{M}$, and solving the system equations where all these minors are equal to zero. While it only involves the x_k variables, the degree of each equation is $r+1$. This causes complexity of the method to rely heavily on the general method of solving system of multivariate equations using Gröbner basis, which has complexity $O(\binom{K+d}{d}^\omega)$ where d is the degree of regularity for the determinant ideal, and ω is the constant for matrix multiplication.

Kipnis–Shamir attack [KS99] tries to solve for the right kernel of $\mathbf{M}$. Since $\mathbf{M}$ is of size m-by-n and has rank at most r, its right kernel has at least $n-r$ dimensions, which means $n-r$ linear independent vectors $\mathbf{y}_k$ can be chosen such that $\mathbf{M}\mathbf{y}_k = \mathbf{0}$. Different with kernel attack, Kipnis–Shamir attack sets new variables as coordinates of $\mathbf{y}_k$'s, and gets bilinear quadratic equations. Kipnis–Shamir attack is analyzed [FLP08] to contain equations in Minors attack. For more information about the complexity of Kipnis–Shamir attack we refer the readers to [FLP08, FDS10, FDS13, VBC$^+$19, WINT20, NWI23].

Support-Minors attack is the state-of-the-art method of solving homogeneous MinRank problems. It decomposes the matrix $\mathbf{M}$ as product of two rank r matrices $\mathbf{M} = \mathbf{SC}$ where $\mathbf{C}$ is a r-by-n matrix, and sets the maximal minors of $\mathbf{C}$ as new variables. Equations are obtained by augmenting $\mathbf{C}$ with each row of $\mathbf{M}$, and letting the new maximal minors (the size increased by one) be zero. This new attack has been analyzed [BB22, GD22] to contain the equations in Kipnis–Shamir attack. For complexity of Support-Minors attack we refer the readers to [BBC$^+$20].

2.4 Previous MinRank attacks on UOV-like schemes

Among UOV-like schemes, MinRank attack was first applied to Rainbow [DS05], where a linear combination of public key matrices has exceptionally small rank. In this attack the matrices are chosen as the public key itself. In [Beu21a] the author introduced a new type of MinRank attack on Rainbow, called *rectangular* MinRank attack. The idea can be abstracted using Ikematsu's matrix deformation [INT23]: Let $(\mathbf{Q}_1, \ldots, \mathbf{Q}_m)$ be a set of n-by-n matrices over $\mathbb{F}_q$, and let $\mathbf{q}_k^{(j)}$ denote the j-th column vector of $\mathbf{Q}_k$. Then we

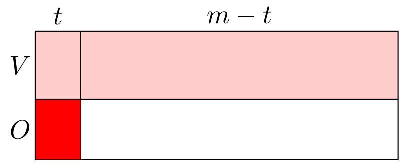


Figure 1: Shape of $\tilde{\mathbf{F}}_N$. The rank does not exceed $V + t$.

define the new set $(\tilde{\mathbf{Q}}_1, \ldots, \tilde{\mathbf{Q}}_n)$ of n-by-m matrices as

$$\begin{aligned} \tilde{\mathbf{Q}}_1 &= \begin{bmatrix} \mathbf{q}_1^{(1)} & \mathbf{q}_2^{(1)} & \cdots & \mathbf{q}_m^{(1)} \end{bmatrix} \\ \tilde{\mathbf{Q}}_2 &= \begin{bmatrix} \mathbf{q}_1^{(2)} & \mathbf{q}_2^{(2)} & \cdots & \mathbf{q}_m^{(2)} \end{bmatrix} \\ &\vdots \\ \tilde{\mathbf{Q}}_n &= \begin{bmatrix} \mathbf{q}_1^{(n)} & \mathbf{q}_2^{(n)} & \cdots & \mathbf{q}_m^{(n)} \end{bmatrix} \end{aligned} \tag{1}$$

It is stated in [INT23] that if $\mathbf{S}$ is an n-by-n matrix and $\mathbf{T}$ is an m-by-m matrix, and $(\mathbf{F}_1, \ldots, \mathbf{F}_m)$ is a set of n-by-n matrices, then the matrix deformation of $(\mathbf{P}_1, \ldots, \mathbf{P}_m) = (\mathbf{S}\mathbf{F}_1\mathbf{S}^t, \ldots, \mathbf{S}\mathbf{F}_m\mathbf{S}^t)\mathbf{T}$ is

$$(\tilde{\mathbf{P}}_1, \ldots, \tilde{\mathbf{P}}_n) = (\mathbf{S}\tilde{\mathbf{F}}_1\mathbf{T}, \ldots, \mathbf{S}\tilde{\mathbf{F}}_n\mathbf{T})\mathbf{S}^t \tag{2}$$

Therefore if some of the $\tilde{\mathbf{F}}_i$'s have some low rank property, then a linear combination of $\tilde{\mathbf{P}}_i$ should also be low rank.

[FI23] also applied rectangular MinRank attack on MAYO [Beu21b] and QR-UOV [FIKT21], and confirmed that MAYO and QR-UOV are secure under rectangular MinRank attack. VOX, however, is shown to be weak under this attack. In [MPC+23], the authors summarized the attack given by [FI23]. The idea is to notice that if we view the UOV map as on extension field $\mathbb{F}_{q^c}$ and generate the $\mathbf{F}_i$'s and $\mathbf{P}_i$'s correspondingly, the matrix deformation $\tilde{\mathbf{F}}_N$ have rank at most $V + t$, due to its special shape: the last $m - t$ columns of $\tilde{\mathbf{F}}_N$ have the last O rows as zero rows, so the rank they can contribute is at most V; the first t columns of $\tilde{\mathbf{F}}_N$ are random, however since $O > t$, the rank they can contribute additionally is at most t.

Since $\mathbf{S}\tilde{\mathbf{F}}_N\mathbf{T}$ is a linear combination of $\tilde{\mathbf{P}}_1, \ldots, \tilde{\mathbf{P}}_N$, this creates a MinRank instance. The authors used the support minors method to estimate the complexity of the attack, and the results are listed in Table 4.

Table 4: Complexity of the Rectangular MinRank attack on VOX parameters

λ	q	$O = m/c$	$V = v/c$	c	t	$\log_2 C$
128	251	8	9	6	6	50.8
192	1021	10	11	7	7	54.8
256	4093	12	13	8	8	55.3

3 A Practical Attack Against VOX

Our first idea comes from the disadvantage that rectangular MinRank attack cannot be applied to VOX, due to the fact that $\tilde{\mathbf{F}}_i$'s are all full row rank now. However, if we concatenate $\tilde{\mathbf{F}}_{N-1}$ and $\tilde{\mathbf{F}}_N$ vertically, the concatenated matrix will have rank at most $2V + t$, due to the fact that $2O > t$ and $m - t > 2V$ for the parameters in Table 3.

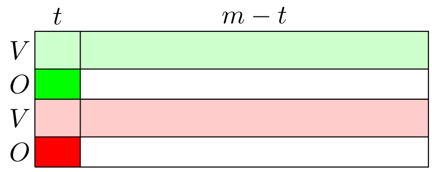


Figure 2: The shape of $\begin{bmatrix}\tilde{\mathbf{F}}_{N-1}\\ \tilde{\mathbf{F}}_N\end{bmatrix}$. The rank does not exceed $2V+t$.

Generally, for $l \le O$, if $m-t > lV$ and $lO > t$, then the following matrix

$$\mathbf{M}'_{\mathbf{s}} = \begin{bmatrix}\mathbf{S}\tilde{\mathbf{F}}_{N-l+1}\mathbf{T}\\ \vdots \\ \mathbf{S}\tilde{\mathbf{F}}_N\mathbf{T}\end{bmatrix} = \begin{bmatrix}\mathbf{S}\tilde{\mathbf{F}}_{N-l+1}\\ \vdots \\ \mathbf{S}\tilde{\mathbf{F}}_N\end{bmatrix}\mathbf{T} = (\mathbf{I}_l \otimes \mathbf{S})\begin{bmatrix}\tilde{\mathbf{F}}_{N-l+1}\\ \vdots \\ \tilde{\mathbf{F}}_N\end{bmatrix}\mathbf{T}$$

has rank at most $lV+t$. Using the formula (2), since $\mathbf{S}\tilde{\mathbf{F}}_{N-l+1}\mathbf{T}, \ldots, \mathbf{S}\tilde{\mathbf{F}}_N\mathbf{T}$ are all linear combinations of $\tilde{\mathbf{P}}_1, \ldots, \tilde{\mathbf{P}}_N$, it seems that we need to find choices of $x_{1,i}, \ldots, x_{l,i}$ such that

$$\mathbf{M}'_{\mathbf{s}} = \begin{bmatrix}\sum_{i=1}^N x_{1,i}\tilde{\mathbf{P}}_i\\ \vdots \\ \sum_{i=1}^N x_{l,i}\tilde{\mathbf{P}}_i\end{bmatrix}$$

has rank at most $lV+t$. However, if we naively solve this, we will get many spurious solutions which we do not really want. For example, if we choose $x_{1,i} = \cdots = x_{l,i}$ for all i, then $\mathbf{M}'_{\mathbf{s}}$ will have rank at most N, which is not what we want. However, from (2) notice that $\mathbf{x}_j = (x_{j,1}, \ldots, x_{j,N})$ should be the $N-l+j$ column of $(\mathbf{S}^t)^{-1}$, which is a block upper triangular matrix, therefore we have $x_{j,i} = \delta_{i,N-l+j}$ for $i > V$. As such we have

$$\mathbf{M}_{\mathbf{s}} = \begin{bmatrix}\sum_{i=1}^V x_{1,i}\tilde{\mathbf{P}}_i + \tilde{\mathbf{P}}_{N-l+1}\\ \vdots \\ \sum_{i=1}^V x_{l,i}\tilde{\mathbf{P}}_i + \tilde{\mathbf{P}}_N\end{bmatrix} \tag{3}$$

which is an inhomogeneous MinRank instance. If we write out each component of linear combination, we notice that each component has the form of

$$\tilde{\mathbf{F}}_i^{(j,l)} = \begin{bmatrix}\mathbf{0}\\ \vdots \\ \tilde{\mathbf{F}}_i \\ \vdots \\ \mathbf{0}\end{bmatrix} \tag{4}$$

where l is the number of matrices concatenated, hence the name “padded” rectangular MinRank.

3.1 Nontrivial rank fall of $\mathbf{M}_{\mathbf{s}}$

In this subsection we show that, due to the symmetry property of public key and central map, the rows of $\mathbf{M}_{\mathbf{s}}$ have a structured linear combination which amounts to zero. Recall that if the central map $\mathbf{F}_i$'s are symmetric, so are the public keys $\mathbf{P}_i$. Now notice that

$$\sum_{i=1}^V x_{1,i}\tilde{\mathbf{P}}_i + \tilde{\mathbf{P}}_{N-l+1} = \begin{bmatrix}\mathbf{P}_1\mathbf{x}_1{}^t & \mathbf{P}_2\mathbf{x}_1{}^t & \cdots & \mathbf{P}_o\mathbf{x}_1{}^t\end{bmatrix}$$

Table 5: Experiment result of our attack.

λ	q	O	V	c	t	Running time (second)	Total Memory Usage (MB)
128	251	4	5	13	6	0.170	32.09
		5	6	11	6	0.510	32.09
		6	7	9	6	27357.799	6147.06
192	1021	5	6	15	7	0.440	32.09
		6	7	13	7	0.790	32.09
		7	8	11	7	26.170	157.69
256	4093	6	7	17	8	1.240	64.12
		7	8	14	8	1.870	64.12
		8	9	13	8	51.530	256.00

Similarly we have

$$\sum_{i=1}^{V} x_{2,i}\tilde{\mathbf{P}}_i + \tilde{\mathbf{P}}_{N-l+2} = \begin{bmatrix}\mathbf{P}_1\mathbf{x}_2{}^t & \mathbf{P}_2\mathbf{x}_2{}^t & \cdots & \mathbf{P}_o\mathbf{x}_2{}^t\end{bmatrix}$$

Therefore

$$\begin{aligned}\mathbf{x}_2\left(\sum_{i=1}^{V} x_{1,i}\tilde{\mathbf{P}}_i + \tilde{\mathbf{P}}_{N-l+1}\right) &= \begin{bmatrix}\mathbf{x}_2\mathbf{P}_1\mathbf{x}_1{}^t & \mathbf{x}_2\mathbf{P}_2\mathbf{x}_1{}^t & \cdots & \mathbf{x}_2\mathbf{P}_o\mathbf{x}_1{}^t\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{x}_1\mathbf{P}_1\mathbf{x}_2{}^t & \mathbf{x}_1\mathbf{P}_2\mathbf{x}_2{}^t & \cdots & \mathbf{x}_1\mathbf{P}_o\mathbf{x}_2{}^t\end{bmatrix}\\ &= \mathbf{x}_1\left(\sum_{i=1}^{V} x_{2,i}\tilde{\mathbf{P}}_i + \tilde{\mathbf{P}}_{N-l+2}\right)\end{aligned}$$

which shows that a nonzero linear combination of the first $2N$ rows is zero. For every pair of blocks such syzygy exists, so we expect $\mathbf{M_s}$ to have rank at most $lN - \binom{l}{2}$. To make the MinRank attack works, the parameters should satisfy $lV + t < lN - \binom{l}{2}$, or equivalently $t < lO - \binom{l}{2}$. Since l can be $1, 2, \ldots, O$, we expect that such attack works when $t < O(O + 1)/2$.

3.2 Experimental results

Since we are dealing with an inhomogeneous MinRank instance, we adapt the Kipnis–Shamir attack and solve for the left kernel of $\mathbf{M_s}$. The equations come from the following matrix equation:

$$\begin{bmatrix}\mathbf{K} & \mathbf{I}_{N-r}\end{bmatrix}\mathbf{M_s} = \mathbf{0} \tag{5}$$

where $\mathbf{K}$ is an $(N - r)$-by-N matrix whose entries form the kernel variables.

To solve for the Gröbner basis of the ideal generated by the Kipnis–Shamir attack, we used the Gröbner basis algorithm F4 with respect to the graded reverse lexicographic monomial order in Magma V2.28-2 [BCP97] on CPU a 2.40GHz Intel Xeon Silver 4214R CPU. The Magma code we use can be viewed in Appendix A and on Github[1]. The detailed running time of the Gröbner basis solving is listed in Table 5.

The attack costs less than one second for the first two parameters of level 1 and level 3, less than two seconds for the first two parameters of level 5, and less than one minute for the other parameters except the slowest one. In the experiment, we saw that all the nine systems have first degree fall at degree 3, which matches the analysis above.

[1] https://github.com/tuovsig/analysis

Table 6: Estimated complexity of our practical attack.

λ	q	$O = m/c$	$V = v/c$	c	t	d	D_{mgd}	$\log_2 C$
128	251	4	5	13	6	1	5	41.28
		5	6	11	6	1	6	49.64
		6	7	9	6	1	7	58.02
192	1021	5	6	15	7	2	4	43.41
		6	7	13	7	1	5	45.92
		7	8	11	7	1	6	54.54
256	4093	6	7	17	8	1	4	45.35
		7	8	14	8	1	5	48.04
		8	9	13	8	2	6	56.83

Table 7: Estimated complexity of our attack on possible VOX parameters.

λ	q	$O = m/c$	$V = v/c$	c	t	d	D_{mgd}	$\log_2 C$
128	251	4	7	13	6	1	8	78.10
		5	9	11	6	1	11	99.80
		6	11	9	6	2	13	133.96
192	1021	5	9	15	7	1	8	84.95
		6	11	13	7	1	10	101.51
		7	13	11	7	1	14	129.55
256	4093	6	11	17	8	1	8	90.60
		7	13	14	8	1	11	113.72
		8	15	13	8	1	13	130.61

3.3 Our hypothetical analysis for the result

To give a theoretical upper bound for the complexity of our attack, here we adopt the analysis of [NWI23], and introduce the monomial graded degree D_{mgd} which is the smallest total degree of monomials in

$$\frac{\prod_{i=1}^{d}(1-t_0t_i)^m}{(1-t_0)^{lV}(1-t_1)^r\dots(1-t_d)^r}$$

whose coefficient is negative. The monomial D_{mgd} is believed to bound from above the solving degree, hence it gives an upper bound for the complexity estimation. d is the number of kernel vectors we choose, and should range between 1 and $lN - r$. Using the formula $\binom{lV+dr+D_{mgd}}{D_{mgd}}^{\omega}$ to estimate the complexity C, we list the complexity estimation in Table 6.

Using this estimation, we try to fix the parameters for VOX. It is hard to tweak t respect to O, because small t will not exceed $lO - \binom{l}{2}$, while large t will make signature harder due to Gröbner basis calculation. While making c smaller can reduce the equations occurred in Kipnis–Shamir method, it will decrease the number of variables when viewed over $\mathbb{F}_q$, resulting in a decrease of security. Therefore we decided to only tweak V. We found that the complexity grows as V increases, and we checked the parameters for $V < 2O$. We found that all of the parameters still fail the estimation, with complexity less than 140 bits.

4 Another Attack Against VOX

Our second idea comes from a flaw in QR-structure of VOX, specifically when parameter c is a composite number, which results in the presence of an intermediate field within the field extension $\mathbb{F}_q \subset \mathbb{F}_{q^c}$ used in the VOX. Consequently, we can consider the public key as a polynomial over this intermediate field, and subsequently construct a matrix that is not full rank.

Recall that in the QR-structure, every $a \in \mathbb{F}_{q^c}$ can be expressed as a $c \times c$ matrix over $\mathbb{F}_q$ [FIKT21, PCF+23]. Specifically, let $g \in \mathbb{F}_{q^c}$ be a root of an irreducible polynomial of degree c over $\mathbb{F}_q$. The matrix expression $\Phi(a)$ is given by the following ring homomorphism:

$$\begin{aligned} \Phi : &\mathbb{F}_{q^c} \hookrightarrow \mathsf{Mat}_c(\mathbb{F}_q) \\ &a \mapsto \Phi(a), \qquad \text{where } (1, g, \ldots, g^{c-1})\Phi(a) = (a, ag, \ldots, ag^{c-1}). \end{aligned}$$

In order to realize this attack, we focus on the case where c is a composite number and can be factored as $c = c_1c_2$, allowing us to express $a \in \mathbb{F}_{q^c}$ as a matrix over an intermediate field. In this case, the matrix expression is given by a ring homomorphism $\Psi : \mathbb{F}_{q^c} \hookrightarrow \mathsf{Mat}_{c_2}(\mathbb{F}_{q^{c_1}})$. The design of Ψ will be detailed in the following section.

Moreover, we can induce a map on matrix ring from Ψ

$$\begin{aligned} \mathsf{Mat}_N(\Psi) : &\mathsf{Mat}_N(\mathbb{F}_{q^c}) \to \mathsf{Mat}_N(\mathsf{Mat}_{c_2}(\mathbb{F}_{q^{c_1}})) = \mathsf{Mat}_{c_2N}(\mathbb{F}_{q^{c_1}}) \\ &(a_{ij})_{N\times N} \mapsto (\Psi(a_{ij}))_{N\times N} \end{aligned}$$

It is straightforward to observe that this is also a ring homomorphism owing to the homomorphic property of Ψ. For matrix $\mathbf{P} \in \mathsf{Mat}_N(\mathbb{F}_{q^c})$, we denote $\mathbf{P}^\Psi \in \mathsf{Mat}_{c_2N}(\mathbb{F}_{q^{c_1}})$ as the image $\mathbf{P}$ under map $\mathsf{Mat}_N(\Psi)$ in the following.

Applying the ring homomorphism $\mathsf{Mat}_N(\Psi)$ to VOX public keys, we have

$$(\mathbf{P}_1^\Psi, \ldots, \mathbf{P}_m^\Psi) = (\mathbf{S}^\Psi\mathbf{F}_1^\Psi\mathbf{S}^{t\Psi}, \ldots, \mathbf{S}^\Psi\mathbf{F}_m^\Psi\mathbf{S}^{t\Psi})\mathbf{T}$$

The matrix deformation of $(\mathbf{P}_1^\Psi, \ldots, \mathbf{P}_m^\Psi)$ (*resp.* $(\mathbf{F}_1^\Psi, \ldots, \mathbf{F}_m^\Psi)$) is denoted as $(\widetilde{\mathbf{P}_1^\Psi}, \ldots, \widetilde{\mathbf{P}_{c_2N}^\Psi})$ (*resp.* $(\widetilde{\mathbf{F}_1^\Psi}, \ldots, \widetilde{\mathbf{F}_{c_2N}^\Psi})$). As (2), we have

$$(\widetilde{\mathbf{P}_1^\Psi}, \ldots, \widetilde{\mathbf{P}_{c_2N}^\Psi}) = (\mathbf{S}^\Psi\widetilde{\mathbf{F}_1^\Psi}\mathbf{T}, \ldots, \mathbf{S}^\Psi\widetilde{\mathbf{F}_{c_2N}^\Psi}\mathbf{T})\mathbf{S}^{t\Psi} \tag{6}$$

We can choose a factor c_2 of c such that $c_2V + t < c_2N$, then the matrices $\widetilde{\mathbf{F}_i^\Psi}$, $i = c_2V + 1, \ldots, c_2N$ have low rank.

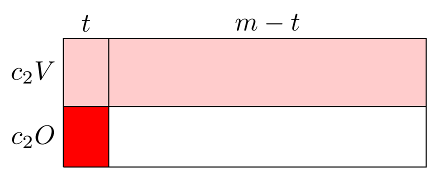


Figure 3: Shape of $\widetilde{\mathbf{F}_{c_2N}^\Psi}$. The rank does not exceed $c_2V + t$.

Generally, for a factor c_2 of c, if $m - t > c_2V$ and $c_2O > t$, then the matrix $\mathbf{M_s} = \mathbf{S}^\Psi\widetilde{\mathbf{F}_{c_2N}^\Psi}\mathbf{T}$ has rank at most $c_2V + t$. Using the formula (6), since $\mathbf{M_s}$ is linear combinations of $\widetilde{\mathbf{P}_1^\Psi}, \ldots, \widetilde{\mathbf{P}_{c_2N}^\Psi}$, it seems that we need to find choices of $x_1, \ldots, x_{c_2N}$ such that $\mathbf{M_s} = \sum_{i=1}^{c_2N} x_i\tilde{\mathbf{P}}_i$ has rank at most $c_2V + t$. From (6) notice that $\mathbf{x} = (x_1, \ldots, x_{c_2N})$ should be the last column of $(\mathbf{S}^{t\Psi})^{-1}$, which is a block upper triangular matrix, therefore we have $x_i = \delta_{i,c_2N}$ for $i > V$.

4.1 Matrix expression over intermediate field

In this section, we show the design of ring homomorphism $\Psi : \mathbb{F}_{q^c} \hookrightarrow \mathsf{Mat}_{c_2}(\mathbb{F}_{q^{c_1}})$ which brings the matrix expression over intermediate field $\mathbb{F}_{q^{c_1}}$ of element in $\mathbb{F}_{q^c}$.

In general, $\mathbb{F}_{q^c}$ is a linear space over intermediate field $\mathbb{F}_{q^{c_1}}$ of dimension c_2. Fix a basis of $\mathbb{F}_{q^c}$, denoted as $(\alpha_0, \alpha_1, \dots, \alpha_{c_2-1})$, then we have a nature ring homomorphism

$$\begin{aligned}\Psi' :& \mathbb{F}_{q^c} \hookrightarrow \mathsf{Mat}_{c_2}(\mathbb{F}_{q^{c_1}})\\ & a \mapsto \Psi(a), \qquad \text{where } (\alpha_0, \alpha_1, \dots, \alpha_{c_2-1})\Psi(a) = (a\alpha_0, a\alpha_1, \dots, a\alpha_{c_2-1})\end{aligned}$$

Specifically, let $g \in \mathbb{F}_{q^c}$ be a root of an irreducible polynomial of degree c over $\mathbb{F}_q$ and $h \in \mathbb{F}_{q^c}$ be a root of an irreducible polynomial of degree c_1 over $\mathbb{F}_q$, then we get the intermediate field $\mathbb{F}_q[h]$ and field extension $\mathbb{F}_q[h][g] = \mathbb{F}_q[g] = \mathbb{F}_{q^c}$. Note that $\mathbb{F}_q[g]$ is a linear space over $\mathbb{F}_q[h]$ with basis $(1, g, \dots, g^{c_2-1})$, then we can construct the ring homomorphism

$$\begin{aligned}\Psi :& \mathbb{F}_{q^c} \hookrightarrow \mathsf{Mat}_{c_2}(\mathbb{F}_q[h])\\ & a \mapsto \Psi(a), \qquad \text{where } (1, g, \dots, g^{c_2-1})\Psi(a) = (a, ag, \dots, ag^{c_2-1})\end{aligned}$$

Every column of matrix $\Psi(a)$ is the coordinates of ag^i under the basis $(1, g, \cdots, g^{c_2-1})$. For every $a = \sum_{i=0}^{c-1} x_i g^i \in \mathbb{F}_q[g]$, we can compute the coordinates easily. Since $(1, g, \dots, g^{c-1})$ and

$$(1, h, \dots, h^{c_1-1}, g, gh, \dots, gh^{c_1-1}, \dots, g^{c_2-1}, g^{c_2-1}h, \dots, g^{c_2-1}h^{c_1-1})$$

form two $\mathbb{F}_q$-bases of $\mathbb{F}_{q^c}$. We set $\mathbf{G}$ as the transition matrix between the two bases. We can also written a as $\sum_{i,j} y_{c_1 i+j} g^i h^j$, where $y_k \in \mathbb{F}_q$, and if we set $\mathbf{x} = (x_0, x_1, \dots, x_{c-1})^t$, $\mathbf{y} = (y_0, y_1, \dots, y_{c-1})^t$, we have $\mathbf{y} = \mathbf{G}\mathbf{x}$. Then we get the coordinate of a under $(1, g, \dots, g^{c-1})$.

$$\begin{aligned} a &= \left(1, g, \cdots, g^{c-1}\right)\mathbf{x}\\ &= \left(1, h, \dots, h^{c_1-1}, g, gh, \dots, gh^{c_1-1}, \dots, g^{c_2-1}, g^{c_2-1}h, \dots, g^{c_2-1}h^{c_1-1}\right)\mathbf{G}\mathbf{x}\\ &= \left(1, g, \cdots, g^{c_2-1}\right)\begin{pmatrix}\sum_{i=0}^{c_1-1} y_i h^i\\ \sum_{i=0}^{c_1-1} y_{i+c_1} h^i\\ \vdots\\ \sum_{i=0}^{c_1-1} y_{i+c_1(c_2-1)} h^i\end{pmatrix}\end{aligned}$$

4.2 Our hypothetical analysis for the result

We adapt the Kipnis-Shamir method for solving MinRank problem. We can estimate the complexity of our attack following the complexity analysis detailed in Section 3.3. The estimated complexity is listed in Table 8.

Table 8: Estimated complexity of MinRank attack over the intermediate field $\mathbb{F}_{q^{c_1}}$ on VOX parameters.

λ	q	$O = m/c$	$V = v/c$	c	c_1	t	d	D_{mgd}	$\log_2 C$
128	251	6	7	9	3	6	2	12	112.46
	251	5	6	10	5	6	1	6	49.64
192	1021	5	6	15	5	7	1	8	69.48
256	4093	7	8	14	7	8	1	5	48.04

To investigate whether the estimated complexity accurately reflects the actual complexity, we experimented for VOX with such a smaller parameter. We used the Gröbner basis algorithm F4 with respect to the graded reverse lexicographic monomial order in

Magma V2.28-2 [BCP97] on CPU a 2.40GHz Intel Xeon Silver 4214R CPU. The Magma code we use can be viewed in Appendix B and on Github[2]. The detailed running time of the Gröbner basis solving is listed in Table 9.

Table 9: Experiment results of MinRank attack over the intermediate field $\mathbb{F}_{q^{c_1}}$ on smaller VOX parameters.

q	$O = m/c$	$V = v/c$	c	c_1	t	d	D_{mgd}	$\log_2 C$	Running time(s)	Memory Usage(MB)
251	4	5	14	7	5	1	4	34.54	6.219	32.09
251	4	5	16	8	5	1	4	34.54	5.750	32.09
251	4	5	14	7	6	1	4	41.14	120.969	86.56
251	5	6	14	7	6	1	5	43.5	769.649	310.62
251	5	6	14	7	6	2	4	42.88	942.580	448.16
251	5	6	16	8	6	1	4	36.83	11.980	32.09
1021	5	6	16	8	7	1	4	37.25	49.789	64.12
1021	5	6	16	8	7	2	4	43.41	1374.059	499.12

From the experimental results, we observe that there is a nearly direct proportional relationship between the logarithm base 2 of running time and the logarithm base 2 of the estimated complexity, which we denote as $\log_2 C$. By applying linear regression, the fitting equation is $y = 1.15x + 31.86$ which has a slope near 1. This suggests that the estimated complexity provides a good prediction of the actual complexity. The fitted line is depicted in Figure 4.

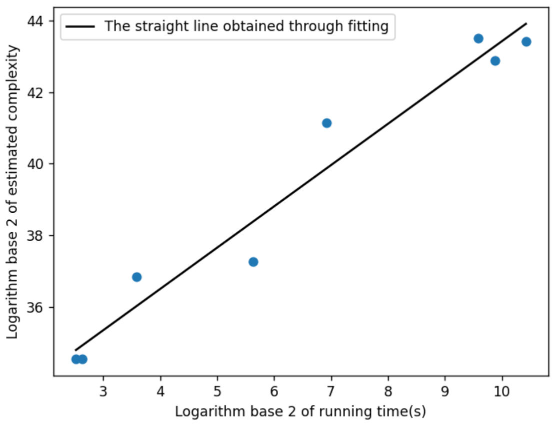


Figure 4: Experiment running time and estimated complexity as well as the related fitted line of MinRank attack over the intermediate field $\mathbb{F}_{q^{c_1}}$ on smaller VOX parameters.

5 Conclusion

This paper presents two MinRank-based attacks against new parameters of VOX scheme, which has been submitted to NIST Post-Quantum Cryptography Project. The first attack pads public matrices vertically, and it can recover most of VOX oil spaces in seconds. While practically powerful, the padding attack lacks theoretic analysis. Hence we introduce another attack that can drastically decrease VOX security level in theory. It constructs intermediate field when "c" is co-prime and experiments on small parameters substantiate the hypothetical analysis.

[2]https://github.com/tuovsig/analysis

With these two attacks breaking VOX in different approaches, we suspect that there might be some unspecified vulnerabilities in the scheme construction that could induce more fundamental security problems. Moreover, we presume that, in the practical attack, the gap between the passable hypothetical analysis and marvelous experiment results comes from the sparseness in matrices. It would be interesting to reason the discrepancy. Last but not the least, we expect that our attacks could be further applied to other UOV-like schemes.

References

[BB22] Magali Bardet and Manon Bertin. Improvement of algebraic attacks for solving superdetermined minrank instances. In Jung Hee Cheon and Thomas Johansson, editors, *Post-Quantum Cryptography - 13th International Workshop, PQCrypto 2022, Virtual Event, September 28-30, 2022, Proceedings*, volume 13512 of *Lecture Notes in Computer Science*, pages 107–123. Springer, 2022.

[BBC+20] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray A. Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier A. Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems. In Shiho Moriai and Huaxiong Wang, editors, *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*, volume 12491 of *Lecture Notes in Computer Science*, pages 507–536. Springer, 2020.

[BCC+23] Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias J. Kannwischer. MAYO. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

[BCD+23] Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang. UOV: Unbalanced Oil and Vinegar - Algorithm Specifications and Supporting Documentation Version 1.0. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

[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).

[Beu21a] Ward Beullens. Improved cryptanalysis of UOV and rainbow. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology - EUROCRYPT 2021 - 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17-21, 2021, Proceedings, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 348–373. Springer, 2021.

[Beu21b] Ward Beullens. MAYO: practical post-quantum signatures from oil-and-vinegar maps. In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers*, volume 13203 of *Lecture Notes in Computer Science*, pages 355–376. Springer, 2021.

[Cou01] Nicolas T. Courtois. Efficient zero-knowledge authentication based on a linear algebra problem minrank. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 402–421. Springer, 2001.

[DGG+23] Jintai Ding, Boru Gong, Hao Guo, Xiaoou He, Yi Jin, Yuansheng Pan, Dieter Schmidt, Chengdong Tao, Danli Xie, Bo-Yin Yang, and Ziyu Zhao. TUOV: Triangular Unbalanced Oil and Vinegar - Algorithm Specifications and Supporting Documentation Version 1.0. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, volume 3531 of *Lecture Notes in Computer Science*, pages 164–175, 2005.

[FDS10] Jean-Charles Faugère, Mohab Safey El Din, and Pierre-Jean Spaenlehauer. Computing loci of rank defects of linear matrices using gröbner bases and applications to cryptology. In Wolfram Koepf, editor, *Symbolic and Algebraic Computation, International Symposium, ISSAC 2010, Munich, Germany, July 25-28, 2010, Proceedings*, pages 257–264. ACM, 2010.

[FDS13] Jean-Charles Faugère, Mohab Safey El Din, and Pierre-Jean Spaenlehauer. On the complexity of the generalized minrank problem. *J. Symb. Comput.*, 55:30–58, 2013.

[FI23] Hiroki Furue and Yasuhiko Ikematsu. A new security analysis against MAYO and QR-UOV using rectangular minrank attack. In Junji Shikata and Hiroki Kuzuno, editors, *Advances in Information and Computer Security - 18th International Workshop on Security, IWSEC 2023, Yokohama, Japan, August 29-31, 2023, Proceedings*, volume 14128 of *Lecture Notes in Computer Science*, pages 101–116. Springer, 2023.

[FIH+23] Hiroki Furue, Yasuhiko Ikematsu, Fumitaka Hoshino, Tsuyoshi Takagi, Kan Yasuda, Toshiyuki Miyazawa, Tsunekazu Saito, and Akira Nagai. QR-UOV. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

[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 - 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6-10, 2021, Proceedings, Part IV*, volume 13093 of *Lecture Notes in Computer Science*, pages 187–217. Springer, 2021.

[FLP08] Jean-Charles Faugère, Françoise Levy-dit-Vehel, and Ludovic Perret. Cryptanalysis of minrank. In David A. Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 280–296. Springer, 2008.

[FmRPP22] Jean-Charles Faugère, Gilles macario Rat, Jacques Patarin, and Ludovic Perret. A New Perturbation for Multivariate Public Key Schemes such as HFE and UOV. Cryptology ePrint Archive, Paper 2022/203, 2022.

[GC00] Louis Goubin and Nicolas T. Courtois. Cryptanalysis of the TTM cryptosystem. In Tatsuaki Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2000.

[GCF+23] Louis Goubin, Benoît Cogliati, Jean-Charles Faugère, Pierre-Alain Fouque, Robin Larrieu, Gilles Macario-Rat, Brice Minaud, and Jacques Patarin. PROV: PRovable unbalanced Oil and Vinegar Specification v1.0 06/01/2023. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

[GD22] Hao Guo and Jintai Ding. Algebraic relation of three minrank algebraic modelings. In Sihem Mesnager and Zhengchun Zhou, editors, *Arithmetic of Finite Fields - 9th International Workshop, WAIFI 2022, Chengdu, China, August 29 - September 2, 2022, Revised Selected Papers*, volume 13638 of *Lecture Notes in Computer Science*, pages 239–249. Springer, 2022.

[INT23] Yasuhiko Ikematsu, Shuhei Nakamura, and Tsuyoshi Takagi. Recent progress in the security evaluation of multivariate public-key cryptography. *IET Inf. Secur.*, 17(2):210–226, 2023.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 1999.

[MPC+23] Gilles Macario-Rat, Jacques Patarin, Benoît Cogliati, Jean-Charles Faugère, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, and Brice Minaud. Rectangular attack on VOX. *IACR Cryptol. ePrint Arch.*, page 1822, 2023.

[NWI23] Shuhei Nakamura, Yacheng Wang, and Yasuhiko Ikematsu. A new analysis of the kipnis-shamir method solving the minrank problem. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, 106(3):203–211, 2023.

[PCF+23] Jacques Patarin, Benoît Cogliati, Jean-Charles Faugère, Pierre-Alain Fouque, Louis Goubin, Robin Larrieu, Gilles Macario-Rat, and Brice Minaud. Vox specification v1.0 06/01/2023. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

[TPD21] Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. Efficient key recovery for all HFE signature variants. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 70–93. Springer, 2021.

[VBC+19] Javier A. Verbel, John Baena, Daniel Cabarcas, Ray A. Perlner, and Daniel Smith-Tone. On the complexity of "superdetermined" minrank instances. In Jintai Ding and Rainer Steinwandt, editors, *Post-Quantum Cryptography -*

10th International Conference, PQCrypto 2019, Chongqing, China, May 8-10, 2019 Revised Selected Papers, volume 11505 of *Lecture Notes in Computer Science*, pages 167–186. Springer, 2019.

[WCD+23] Lih-Chung Wang, Chun-Yen Chou, Jintai Ding, Yen-Liang Kuan, Ming-Siou Li, Bo-Shu Tseng, Po-En Tseng, and Chia-Chun Wang. SNOVA - Proposal for NISTPQC: Digital Signature Schemes project. *Round 1 Additional Signatures, Post-Quantum Cryptography: Digital Signature Schemes*, 2023.

[WINT20] Yacheng Wang, Yasuhiko Ikematsu, Shuhei Nakamura, and Tsuyoshi Takagi. Revisiting the minrank problem on multivariate cryptography. In Ilsun You, editor, *Information Security Applications - 21st International Conference, WISA 2020, Jeju Island, South Korea, August 26-28, 2020, Revised Selected Papers*, volume 12583 of *Lecture Notes in Computer Science*, pages 291–307. Springer, 2020.

A Magma code for our practical attack

Here we list the Magma code we used in Section 3.

```
// parameters for VOX
q := 251;
O := 6;
V := 7;
c := 9;
t := 6;
o := O*c;
v := V*c;
N := O+V;
n := N*c;
m := o;
l := 2;
r := l*V+t;
field<z> := GF(q^c);

// Generation of central map
F0 := [RandomMatrix(field, N, N): i in [1..t]];
F1 := [RandomMatrix(field, V, V): i in [1..m-t]];
F2 := [RandomMatrix(field, V, O): i in [1..m-t]];
F3 := [RandomMatrix(field, O, V): i in [1..m-t]];
FF := F0 cat [VerticalJoin(
    HorizontalJoin(F1[i], F2[i]),
    HorizontalJoin(F3[i], ZeroMatrix(field, O, O))
): i in [1..m-t]];

// Generation of linear map
S2 := RandomMatrix(field, V, O);
S := VerticalJoin(
    HorizontalJoin(
        ScalarMatrix(V, One(field)), S2
    ),
    HorizontalJoin(
        ZeroMatrix(field, O, V), ScalarMatrix(O, One(field))
```

```
    )
);
T2 := RandomMatrix(BaseField(field), t, m-t);
T := VerticalJoin(
    HorizontalJoin(
        ScalarMatrix(t, One(BaseField(field))), T2
    ),
    HorizontalJoin(
        ZeroMatrix(BaseField(field), m-t, t),
        ScalarMatrix(m-t, One(BaseField(field)))
    )
);

// Generation of public key
P := [Transpose(S)*FF[i]*S: i in [1..m]];
PP := [
    &+[T[i][j]*P[j]: j in [1..m]]
    : i in [1..m]
];
PTP := [(Transpose(PP[i]) + PP[i]): i in [1..m]];
PMD := [(Matrix(
    [PTP[j][i]: j in [1..m]]
)): i in [1..N]];

Z := ZeroMatrix(field, m, N*l);
RM := [
    [InsertBlock(Z, PMD[i], 1, N*j+1): i in [1..N]]: j in [0..l-1]
];

// The answer matrix for check
Ans := &+[
    &+[
        -S2[j][O-l+i] * RM[i][j]: j in [1..V]
    ]: i in [1..l]
]
+
&+[
    RM[i][i+N-l]: i in [1..l]
];

// Polynomial Ring, linear variables and kernel variables
PP<[w]> := PolynomialRing(field, l*V+r*(l*N-r), "glex");
X := [Eltseq(w)[(i-1)*V+1..i*V]: i in [1..l]];
Y := [Eltseq(w)[l*V+(i-1)*r+1..l*V+i*r]: i in [1..l*N-r]];

// Matrix M_s and Kernel matrix
MatX := &+[
    &+[
        X[i][j] * RMatrixSpace(PP, m, N*l)!RM[i][j]: j in [1..V]
    ]: i in [1..l]
]
+
```

```
&+[
    RMatrixSpace(PP, m, N*l)!RM[i][i+N-l]: i in [1..l]
];
MatY := Matrix([
    Y[i][1..r] cat [0: j in [r+1..l*N]]: i in [1..l*N-r]
]);
for i in [1..l*N-r] do
    MatY[i][r+i] := 1;
end for;
MatY := Transpose(MatY);

// Generation of equations
KS := MatX * MatY;
Poly := &cat[&cat[[KS[i][j]: j in [1..l*N-r]]: i in [1..m]]];
I := ideal<PP | Poly>;

// Calculate Groebner basis
SetVerbose("Groebner", 1);
time Groebner(I);
print("");
I;
```

B Magma code for our theoretical attack

Here we list the Magma code we used in Section 4.

```
q := 251;
O := 4;
V := 5;
c := 14;
t := 5;
// set d for ks model
d := 1;

o := O*c;
v := V*c;
N := O+V;
n := N*c;
m := o;

c1 := 1;
c2 := c;

for i in [2 .. c] do
    if c mod i eq 0 then
c1 := Round(c/i);
c2 := i;
        break;
    end if;
end for;
```

```
r := c2*V+t;
colstokeep := Minimum(d, c2*N-r);

Fq := GF(q);
R<x> := PolynomialRing(Fq);
f := IrreduciblePolynomial(Fq, c);
fi := IrreduciblePolynomial(Fq, c1);
field<g> := ext< Fq | f >;
interfield<h> := ext< Fq | fi>;
roots := Roots(fi, field);
mu := roots[1][1];
//print(mu);

function EletoMat(a)
    return Transpose(
        Matrix([ElementToSequence(a*g^i): i in [0..c-1]])
    );
end function;

function MattoEle(A)
    return &+ [A[i][1]*h^(i-1) : i in [1..Nrows(A)]];
end function;

function EletoIntermat(a)
    PHIa := EletoMat(a);
    M := Transpose(Matrix(
        &cat [
            [
                ElementToSequence(g^i*mu^j): j in [0..c1-1]
            ]: i in [0..c2-1]
        ]
    ));
    PSIa := M^-1*PHIa*M;
    return Matrix(
        [[MattoEle(
            Submatrix(PSIa,[c1*(i-1)+1..c1*i],[c1*(j-1)+1..c1*j])
        ) : j in [1..c2]] : i in [1..c2]]
    );
end function;

function MatInter(A)
    return VerticalJoin(
        [HorizontalJoin(
            [ EletoIntermat(A[i][j]) : j in [1..Ncols(A)] ]
        ) : i in [1..Nrows(A)]]
    );
end function;

F0 := [RandomMatrix(field, N, N): i in [1..t]];
F1 := [RandomMatrix(field, V, V): i in [1..m-t]];
F2 := [RandomMatrix(field, V, O): i in [1..m-t]];
F3 := [RandomMatrix(field, O, V): i in [1..m-t]];
```

```
FF := F0 cat [VerticalJoin(
    HorizontalJoin(F1[i], F2[i]),
    HorizontalJoin(F3[i], ZeroMatrix(field, O, O))
): i in [1..m-t]];

S2 := RandomMatrix(field, V, O);
S := VerticalJoin(
    HorizontalJoin(
        ScalarMatrix(V, One(field)), S2
    ),
    HorizontalJoin(
        ZeroMatrix(field, O, V), ScalarMatrix(O, One(field))
    )
);

T2 := RandomMatrix(BaseField(field), t, m-t);
T := VerticalJoin(
    HorizontalJoin(
        ScalarMatrix(t, One(BaseField(field))), T2
    ),
    HorizontalJoin(
        ZeroMatrix(
            BaseField(field), m-t, t),
            ScalarMatrix(m-t, One(BaseField(field))
        )
    )
);

P := [Transpose(S)*FF[i]*S: i in [1..m]];
PP := [
    &+[T[i][j]*P[j]: j in [1..m]]
    : i in [1..m]
];

PTP := [(Transpose(PP[i]) + PP[i]): i in [1..m]];

PTPInter := [MatInter(PTP[i]): i in [1..m]];

PMD := [(Matrix(
    [PTPInter[j][i]: j in [1..m]]
)): i in [1..c2*N]];

result := MatInter(Transpose(S^-1))[c2*V+1];

PR<[w]> := PolynomialRing(interfield, c2*V+r*colstokeep, "glex");
X := Eltseq(w)[1..c2*V];
Y := [Eltseq(w)[c2*V+(i-1)*r+1..c2*V+i*r]: i in [1..colstokeep]];
//print(Y);

MatX := &+[X[i]*ChangeRing(PMD[i], PR): i in [1..c2*V]] + PMD[c2*V+1];
```

```
MatY := Matrix([
    Y[i][1..r] cat [0: j in [r+1..c2*N]]: i in [1..colstokeep]
]);
for i in [1..colstokeep] do
    MatY[i][r+i] := 1;
end for;
MatY := Transpose(MatY);

KS := MatX * MatY;
//print(KS);

Poly := &cat[&cat[[KS[i][j]: j in [1..colstokeep]]: i in [1..m]]];
//print(Poly);


I := ideal<PR | Poly>;
//print(result);
SetVerbose("Groebner", 1);
time Groebner(I);
print("");
I;
```