Breaking the Spell: Cryptanalysis of VDOO

Max Cartor[1], Ryann Cartor[2], and Antsa Pierrottet[2]

[1] Department of Mathematics, Bellarmine University, Louisville, Kentucky, USA, mcartor01@bellarmine.edu
[2] School of Mathematical and Statistical Sciences, Clemson University, Clemson, South Carolina, USA

Abstract. VDOO is a Rainbow-like multivariate signature scheme with additional "diagonal" structure intended to resist structural attacks. We show that its security is governed primarily by the first two oil layers, while the diagonal structure has negligible impact, and that all proposed parameters fail to meet their claimed NIST security levels. We provide a corrected complexity analysis of the Simple and Rectangular MinRank attacks, introduce a hybrid Simple–Padded MinRank attack, and compare their effectiveness. Open-source code is provided for independent validation, clarifying VDOO's true security and informing future Rainbow-type post-quantum designs.

Keywords: Multivariate Cryptography · VDOO · Cryptanalysis · MinRank · Rectangular MinRank · Simple Attack · Padded MinRank

1 Introduction

As widespread quantum computing becomes a genuine threat to the classical cryptographic landscape, the need for accurate analysis of quantum-resistant cryptosystems is of the utmost importance. Multivariate cryptography is a possible path forward in the field of post-quantum cryptography. Multivariate schemes typically have quite fast verification/encryption times, and relatively short ciphertexts/signatures in comparison to other post-quantum schemes, such as lattice-based or code-based schemes. However, this efficiency does come with the drawback of increased signing/decryption times and vulnerabilities against rank-based attacks.

The National Institute of Standards and Technology (NIST) has been leading the effort to standardize Post-Quantum cryptosystems. In addition to the newly standardized Post-Quantum Cryptosystems, NIST put out an additional call for signature schemes whose security does not rely on structured lattices, which makes multivariate signature schemes of particular interest. The second round of the Additional Digital Signatures call for the Post-Quantum Cryptography Standardization Process includes the multivariate schemes UOV [21], SNOVA [27], QR-UOV[13] and MAYO[26]—as well as MiRath [1], which is based on the MinRank problem, a central hardness assumption in multivariate cryptography.

The Rainbow signature scheme of [9] was the only multivariate scheme listed as a finalist in Round 3 of the original standardization process, but was eventually dropped after the substantial attacks faced by the Simple Attack of [6]. This emphasizes the need for accurate cryptanalysis of other multivariate cryptosystems, especially those that are a part of the UOV/Rainbow family.

VDOO, proposed by Ganguly, Karmakar, and Saxena in [14], is a multivariate signature scheme inspired by Rainbow. It was designed in hopes of producing a signature scheme that can resist the Simple Attack of [6] while still having practical parameter sets. Evaluating such non-standardized designs is essential: it helps identify structural limitations, improves future constructions, and strengthens our understanding of the multivariate landscape. Both the careful application of existing attacks and the development of new techniques remain crucial for establishing reliable security estimates.

1.1 Our Contribution

In this paper, we present the following contributions.

- In Section 5.5 we introduce a new hybrid attack, combining the Simple Attack [6] and Padded MinRank [16]. We also apply this new attack to VDOO in Section 6.3. This cryptanalysis shows another attack against VDOO that puts its proposed parameters below NIST security levels.
- We provide a thorough and correct analysis of the effects of the Simple Attack and Rectangular MinRank Attack of [6] against the VDOO cryptosystem in Section 6.2. This analysis results in the discovery that parameter sets provided in VDOO, which were thought to be secure, fall below the necessary security requirements.
- We provide implementation code enabling independent verification and experimentation with these attacks, available here.
- In Section 8, we present a reduction that transforms an instance of VDOO into an instance of Rainbow. This enables a direct comparison between the two schemes and facilitates the application (and complexity analysis) of any attack against Rainbow to VDOO.

2 Preliminaries

2.1 Notation

Let q be a prime power and $\mathbb{F}_q$ the finite field with q elements. We use lowercase to denote field elements and bold lowercase to denote vectors. We use calligraphic letters for maps and uppercase bold letters for matrices.

2.2 Solving a system of multivariate polynomials

The security of multivariate cryptosystems is based on the NP-hard Multivariate Quadratic (MQ) problem. The problem is described as follows.

Problem 1 (Multivariate Quadratic (MQ)) *Given a multivariate quadratic map* $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ *and a target vector* $\mathbf{t} \in \mathbb{F}_q^m$*, find a solution* $\mathbf{s}$ *such that* $\mathcal{P}(\mathbf{s}) = \mathbf{t}$.

Note that any system of multivariate polynomials of any degree can be written as a system of multivariate quadratic polynomials by introducing new variables.

The MQ problem is NP-hard, and the best known algorithms for solving it often rely on Gröbner basis algorithms. The MQ problem is believed to be exponentially hard on average, even for quantum computers [4].

Instances of the MQ problem are typically solved using Gröbner basis algorithms, which transform the corresponding polynomial ideal into a form from which the solutions can be readily determined.

Linear algebra based algorithms such as F4 [10] are the fastest known algorithms to compute a degree reverse lexicographical Gröbner basis. These algorithms transform the problem of computing a Gröbner basis into one or more instances of Gaussian elimination of Macaulay matrices. The complexities of these algorithms is dominated by the Gaussian elimination required on the largest Macaulay matrix encountered during the process. The size of the Macaulay matrix is dependent on the degree of the input polynomials, the number N of variables, and on the degree d of the Macaulay matrix.

The Macaulay matrix $\mathcal{M}_d$ of degree d of $\mathcal{F}$ has columns indexed by the monic monomials up to degree d.The rows are indexed by $m_{i,j}f_i$, where $m_{i,j}$ is a monomial such that $\deg(m_{i,j}f_j) \leq d$. The (i,j) entry of $\mathcal{M}_d$ is the coefficient of the monomial of column j in the polynomial corresponding to the i^{th} row.

The complexity of this technique is dependent on the solving degree $d_{\text{sol}}(\mathcal{F})$ of $\mathcal{F}$, which is defined as the least degree d such that Gaussian elimination on the Macaulay matrix $\mathcal{M}_d$ produces a degree reverse lexicographic Gröbner basis of $\mathcal{F}$. To solve, one must compute the Macaulay matrices up to degree $d_{\text{sol}}(\mathcal{F})$, where the largest one encountered has $\sum_{i=1}^{r} \binom{N+d_{\text{sol}}(\mathcal{F})-d_i}{d_{\text{sol}}(\mathcal{F})-d_i}$ many rows and $\binom{N+d_{\text{sol}}(\mathcal{F})}{d_{\text{sol}}(\mathcal{F})}$ many columns, where $d_i = \deg(f_i)$. The complexity of this technique can then be bounded above by

$$\mathcal{O}\left(\binom{N + d_{\text{sol}}(\mathcal{F})}{d_{\text{sol}}(\mathcal{F})}^{\omega}\right)$$

where $2 \leq \omega \leq 3$ is the linear algebra constant.

Another technique to solve a system of equations is to use XL algorithms [8]. Solving a system with the XL algorithm relies on multiplying and linearizing the equations to form and solve an overdetermined linear system. This differs from the Gröbner basis algorithms which systematically perform polynomial reductions to compute a canonical generating set of the ideal.

3 Relevant Schemes

We will discuss the multivariate cryptosystems UOV, Rainbow, VOX and VDOO. These schemes all share a "butterfly" construction, where an easily invertible

quadratic map $\mathcal{F}$ (which we call the "central map") has its structure obscured by composition with invertible linear maps $\mathcal{T}$ and $\mathcal{U}$. The public key $\mathcal{P}(x) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is formed by computing $\mathcal{P} := \mathcal{T} \circ \mathcal{F} \circ \mathcal{U}$. The goal of this composition of functions is to make $\mathcal{P}$ appear to act as a random multivariate quadratic map in the hopes of making it difficult to find a pre-image of $\mathcal{P}(\mathbf{x})$.

3.1 Unbalanced Oil and Vinegar (UOV)

The Oil and Vinegar signature scheme was introduced by Patarin in 1997 [24]. Oil and Vinegar uses two types of variables, namely oil variables and vinegar variables, over a finite field $\mathbb{F}_q$. When the system was first introduced, the number of oil and vinegar variables were equal, but this balanced version of Oil and Vinegar was broken by Kipnis and Shamir in [20]. We now only consider the Unbalanced Oil and Vinegar scheme (UOV), introduced in [19], where the number of vinegar variables is large enough compared to the number of oil variables to be secure against the statistical attacks of [19] and intersection attack of [5].

Let $\mathbf{x} = (x_1, \ldots, x_v, x_{v+1}, \ldots, x_n) \in \mathbb{F}_q^n$. We will call $x_1, \ldots, x_v$ the *vinegar variables*, whereas $x_{v+1}, \ldots, x_n$ will denote the *oil variables*. We define the following central map $\mathcal{F} = (f_1, \ldots, f_{v+1})$, where each f is of the form:

$$f(\mathbf{x}) = \sum_{i=1}^{v} \sum_{j=i}^{v} \alpha_{ij} x_i x_j + \sum_{i=1}^{v} \sum_{j=v+1}^{n} \beta_{ij} x_i x_j, \quad \alpha_{ij}, \beta_{ij} \in \mathbb{F}_q.$$

To create the public key equations $\mathcal{P}$, we compose the linear maps $\mathcal{F}$ with invertible linear maps $\mathcal{U} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ and $\mathcal{T} : \mathbb{F}_q^m \to \mathbb{F}_q^m$, to get $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{U}$.

To sign a message $\mathbf{y}$, the owner of the public key will first compute $\mathbf{z} := \mathcal{T}^{-1}(\mathbf{y})$. Although $\mathcal{F}$ is a quadratic map, $\mathcal{F}$ is linear in the oil variables, which allows the user to easily find a preimage of $\mathbf{z}$. Inversion of the central map is completed by choosing random values in $\mathbb{F}_q$ for each of the vinegar variables, and then solving the remaining linear system for the unknown oil variables. If no solution is found, the signer chooses different values for the vinegar variables and repeats this process until a solution is found. The signer then applies $\mathcal{U}^{-1}$ to the preimage of $\mathbf{z}$ to compute the final signature.

3.2 Rainbow

The Rainbow signature scheme, introduced in [9], can be thought of as a banded construction of UOV consisting of L many UOV layers. Rainbow was the only multivariate signature scheme to make it into the third round of the NIST standardization process, but the scheme was taken out of consideration after facing substantial attacks in [5] and [6].

To create a Rainbow signature scheme, we will still consider input vectors of the form $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ and invertible linear transformations $\mathcal{U} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ and $\mathcal{T} : \mathbb{F}_q^m \to \mathbb{F}_q^m$. The central map $\mathcal{F}$ will now contain L many layers $\mathcal{F} = (F_1, \ldots, F_L)$. The ℓth layer will contain v_ℓ many vinegar variables where $0 < v_1 < v_2 < \ldots < v_L < n$.

Each F_ℓ is composed of $n - v_\ell = o_\ell$ equations. The kth polynomial of F_ℓ will be of the form

$$f_k = \sum_{i=1}^{v_\ell} \sum_{j=1}^{v_\ell} \alpha_{ijk} x_i x_j + \sum_{i=1}^{v_\ell} \sum_{j=(v_\ell)+1}^{v_{\ell+1}} \beta_{ijk} x_i x_j, \qquad \alpha_{ij}, \beta_{ij} \in \mathbb{F}_q.$$

When we let V_ℓ and O_ℓ denote the sets of vinegar variables and oil variables, respectively, in the ℓth layer, we have the nested relationships $V_1 \subset V_2 \subset \cdots \subset V_L$, and $O_L \subset \cdots \subset O_2 \subset O_1$.

As in the case of UOV, the public key is obtained by composing the central map with a linear map $\mathcal{U}$ to get $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{U}$. Note that in the NIST submission, $L = 2$.

To sign a message, the signer first chooses values for the first layer vinegar variables $x_1, \ldots, x_{v_1}$ and substitutes these values into the equations in the first layer F_1. We then solve the resulting linear system in the first layer oil variables $x_{v_1+1}, ..., x_{v_2}$. We next plug the chosen and found values $x_1, \ldots, x_{v_2}$ (now considered our layer 2 vinegar variables) into the layer 2 central maps to solve for the layer 2 oil variables. We continue this process until the L^{th} layer.

3.3 VDOO

The VDOO (Vinegar-Diagonal-Oil-Oil) cryptosystem introduced in [14] has similar construction to Rainbow, but adds a new layer of maps (called the *diagonal layer*) in addition to two Rainbow layers to the central map. The diagonal layer of the central map is composed of d many polynomials, and the ith diagonal polynomial will contain $v + i$ many variables.

We consider the parameters (q, v, d, o_1, o_2) where v is the number of the first batch of vinegar variables, d is the number of diagonal variables, and o_1 and o_2 are the number of the first and the second layer oil variables. We let $n = v + d + o_1 + o_2$ and $m = d + o_1 + o_2$. We denote the central map as $\mathcal{F} = (F_D, F_{O_1}, F_{O_2}) = (f_1, \ldots, f_m)$, where F_D contains d many polynomials and represents the Diagonal Layer, F_{O_1} contains o_1 many polynomials and represents the First Oil Layer, and F_{O_2} contains o_2 many polynomials and represents the Second Oil Layer. The equations in each layer of the central map will have the following forms:

Diagonal Layer. In the diagonal layer, for each $k \in \{v + 1, \ldots, v + d\}$, the polynomial f_{k-v} is defined as follows.

$$f_{k-v}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{k-1} \alpha_{i,k}^{(k)} x_i x_k + \sum_{i,j=1, i \le j}^{k-1} \beta_{i,j}^{(k)} x_i x_j,$$

where $\alpha_{ij}^{(k)}, \beta_{ij}^{(k)} \in \mathbb{F}_q$.

First Oil Layer. In this layer, VDOO uses $(x_1, \ldots, x_{v+d})$ as vinegar variables and considers $(x_{v+d+1}, \ldots x_{v+d+o_1})$ as oil variables. These variables are used to construct o_1 many homogeneous quadratic polynomials of the following form.

$$f_{k-v}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{v+d} \sum_{j=1}^{v+d} \alpha_{ij}^{(k)} x_i x_j + \sum_{i=1}^{v+d} \sum_{j=v+d+1}^{v+d+o_1} \beta_{ij}^{(k)} x_i x_j$$

where $k \in \{v+d+1, \ldots, v+d+o_1\}$, and $\alpha_{ij}^{(k)}, \beta_{ij}^{(k)} \in \mathbb{F}_q$.

Second Oil Layer. The 2nd Oil Layer uses $v+d+o_1$ vinegar variables, o_2 oil variables, and is composed of o_2 quadratic polynomials of the form

$$f_{k-v}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{v+d+o_1} \sum_{j=1}^{v+d+o_1} \alpha_{ij}^{(k)} x_i x_j + \sum_{i=1}^{v+d+o_1} \sum_{j=v+d+o_1+1}^{v+d+o_1+o_2} \beta_{ij}^{(k)} x_i x_j,$$

where $k \in \{v+d+o_1+1, \ldots, v+d+o_1+o_2 = n\}$ and $\alpha_{ij}^{(k)}, \beta_{ij}^{(k)} \in \mathbb{F}_q$.

We again define the public key by composing $\mathcal{F}$ with an invertible linear map $\mathcal{T}$ and $\mathcal{U}$ to get $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{U}$.

Matrix Representation of VDOO Central Maps Consider the symmetric matrix representation of each of the m many VDOO central maps $\mathbf{F}_i \in \mathbb{F}_q^{n \times n}$ such that

$$f_i(x_1, \ldots, x_n) = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix} \mathbf{F}_i \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}^\top.$$

When $i \in \{1, \ldots, d\}$, the matrices $\mathbf{F}_i$ correspond to the diagonal layer maps which have nonzero elements in the upper $(v+i-1) \times (v+i-1)$ block (corresponding to the quadratic vinegar terms), and in the first $(v+i-1)$ coordinates of the $(v+i)$th row and $(v+i)$th column (corresponding to the vinegar terms times the ith diagonal variable).

When $i \in \{d+1, \ldots, d+o_1\}$, the matrices $\mathbf{F}_i$ correspond to the first Oil Layer maps. These $\mathbf{F}_i$ will have guaranteed zeros in the lower right $(o_1+o_2) \times (o_1+o_2)$ block as this corresponds to the quadratic oil terms in the first Oil Layer. These $\mathbf{F}_i$ matrices will also have guaranteed zeros in the last o_2 rows and last o_2 columns, as the first layer oil maps do not include the variables $x_{v+d+o_1+1}, \ldots, x_{v+d+o_1+o_2}$.

Finally, when $i \in \{d+o_1+1, \ldots, d+o_1+o_2\}$, the matrices $\mathbf{F}_i$ correspond to the second Oil Layer maps. These maps $\mathbf{F}_i$ will have guaranteed zeros only in the lower right $o_2 \times o_2$ block as this corresponds to the quadratic terms in the o_2 variables.

In general, the central maps of VDOO will look as shown in Figure 1. If we were to stack all of the matrix representations of the central maps, we can see that the central maps will have the form in Figure 2 where the empty space represents the location of guaranteed zeros.

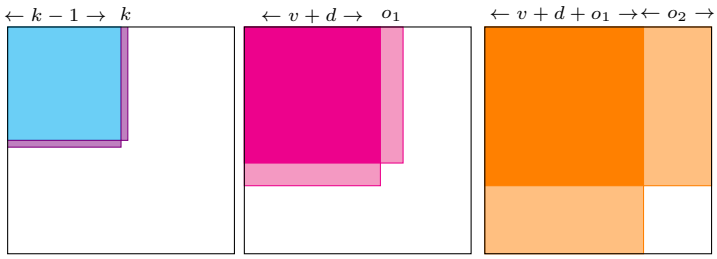


Fig. 1. Visual representation of matrices in the central map. The leftmost image denotes the shape of the first d many equations, f_{k-v} as $k \in [v+1, \ldots, v+d]$ (the diagonal layer). The next two images portray the shape of the equations found in the two UOV layers. The filled-in section of the image denotes possible non-zero terms, whereas the unfilled portions denote guaranteed zeros in the matrices.

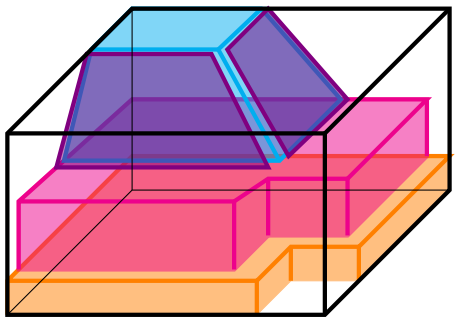


Fig. 2. Stacked matrices of the central map where the color represents possible locations of nonzero terms in each matrix.

4 MinRank

It has been shown that many multivariate cryptosystems are vulnerable to MinRank style attacks. We define the MinRank problem as follows:

Problem 2 (MinRank) *Given K many matrices $\mathbf{A}_1, \ldots, \mathbf{A}_K \in \mathbb{F}_q^{M \times N}$, and $R \in \mathbb{N}$, find coefficients $y_1, \ldots, y_K \in \mathbb{F}_q$ (not all zero) such that*

$$Rank\left(\sum_{i=1}^{K} y_i \mathbf{A}_i\right) \leq R.$$

The goal of MinRank attacks is to find a low rank linear combination of the public matrices. This is an effective attack against many multivariate schemes as it allows an adversary to gain information about the low rank central maps given access only to the public maps. Although the MinRank problem is known to be NP-complete, there are algorithms to solve the problem efficiently enough for some parameter sets of multivariate cryptosystems to be insecure.

The MinRank attack was first introduced in [18] as the first effective attack on HFE [23], which had lead to many new versions of the attack on other cryptosystems with any rank deficiencies. These updated MinRank attacks include Minors Modeling [11], Support Minors Modeling [2], Rectangular MinRank [6], and the Padded MinRank [16].

Multiple techniques have been introduced to solve the MinRank problem. The simplest form is exhaustive search. In this model, one simply guesses linear combinations and computes the rank until a solution is found. The complexity of this version of MinRank is $q^K N^\omega$, where $2 \leq \omega \leq 3$ is the linear algebra exponent.

A simple improvement to exhaustive search is called the combinatorial method introduced in [15], or sometimes linear algebra search. In this method, one guesses $\lceil \frac{K}{M} \rceil$ kernel vectors $\mathbf{v}_i$ and solves

$$\sum_{j=1}^{K} x_j \mathbf{A}_j \mathbf{v}_i = \mathbf{0}$$

linearly for $\mathbf{x}$. Simultaneously guessing two kernel vectors of a rank R matrix takes approximately $q^{\lceil \frac{K}{M} \rceil R}$ attempts on average. Thus, this technique has complexity

$$\mathcal{O}\left(q^{\lceil \frac{K}{M} \rceil R}\left(K^\omega + N^\omega\right)\right).$$

The Support Minors method of MinRank, see [2], is built from decomposition modeling. Given that the matrix $\mathbf{M} = \sum_{i=1}^{K} x_i \mathbf{A}_i$ has rank R, there exist matrices $\mathbf{S} \in \mathbb{F}_q^{M \times R}$ and $\mathbf{C} \in \mathbb{F}_q^{R \times N}$ such that $\mathbf{M} = \mathbf{SC}$. Specifically, the rowspace of $\mathbf{C}$ is the rowspace of $\mathbf{M}$, and so appending a row π of $\mathbf{M}$ to $\mathbf{C}$ results in an $(R+1) \times N$ matrix of rank R. The support minors modeling computes the maximal minors of such a matrix and solves the system for $\mathbf{x}$.

The complexity of solving the MinRank problem using Support Minors Modeling given K many matrices, M equations, N variables, and a target rank of R can be estimated as

$$\min_{M' \leq M} 3(K-1)(R+1)\binom{M'}{R}^2\binom{K+b-2}{b}^2 \tag{1}$$

where b is the smallest integer such that

$$\binom{M'}{R}\binom{K+b-1}{b} - 1 \leq \sum_{i=1}^{b}(-1)^{i+1}\binom{M'}{R+i}\binom{N+i-1}{i}\binom{K+b-i-1}{b-i}. \tag{2}$$

As indicated in Equation (1) above, the number of columns used can be optimized to lower the complexity of the attack. One often finds that even a larger value of b along with a smaller M' may result in a more efficient attack.

Which MinRank technique is best in which parameter regime is a very interesting and subtle question. In general, smaller field sizes paired with either small target rank or a small search space offer more opportunities for the combinatorial search methods to be superior. If the field size is larger, or if the target rank is moderate, Support Minors is often more powerful.

5 Relevant Attacks

5.1 Rectangular MinRank

The Rectangular MinRank attack was first introduced in [6] against Rainbow. This attack introduces a new MinRank instance based on the layered structure of Rainbow with the specific shape of its central map. Later, the Rectangular MinRank attack was abstracted in [17] with the concept of matrix deformation.

This attack makes use of polar forms and symmetric matrix representations of polynomial maps. Given a polynomial $P : \mathbb{F}_q^n \to \mathbb{F}_q^m$, the *polar form* (or *discrete differential*) of P is defined as $DP(\mathbf{x}, \mathbf{y}) := P(\mathbf{x}, \mathbf{y}) - P(\mathbf{x}) - P(\mathbf{y}) - P(\mathbf{0})$. Given the quadratic homogeneous map P, P can be written as $P(\mathbf{x}) = \mathbf{x}\mathbf{P}\mathbf{x}^\top$ for some symmetric matrix $\mathbf{P}$. We call $\mathbf{P}$ the *matrix representation* of P.

Matrix deformation We now discuss the deformation described in [17], used to explain the Rectangular attack from [6]. Let $(\mathbf{G}_1, \ldots, \mathbf{G}_m)$ be a set of $n \times n$ matrices over $\mathbb{F}_q$ and $\mathbf{g}_i^{(j)}$ the j-th column vector of $\mathbf{G}_i$. So $\mathbf{G}_i = \left(\mathbf{g}_i^{(1)}, \ldots, \mathbf{g}_i^{(n)}\right) \in \mathbf{M}_n(\mathbb{F}_q) \in \mathbb{F}_q^{n \times n}$.

Given m many $\mathbf{G}_i$ matrices, we define the *deformation* of $(\mathbf{G}_1, \ldots, \mathbf{G}_m)$ as the new set of matrices $(\widetilde{\mathbf{G}}_1, \ldots, \widetilde{\mathbf{G}}_n)$ such that

$$\begin{aligned}
\widetilde{\mathbf{G}}_1 &= \left(\mathbf{g}_1^{(1)}, \mathbf{g}_2^{(1)}, \ldots, \mathbf{g}_m^{(1)}\right) \\
\widetilde{\mathbf{G}}_2 &= \left(\mathbf{g}_1^{(2)}, \mathbf{g}_2^{(2)}, \ldots, \mathbf{g}_m^{(2)}\right) \\
&\vdots \\
\widetilde{\mathbf{G}}_n &= \left(\mathbf{g}_1^{(n)}, \mathbf{g}_2^{(n)}, \ldots, \mathbf{g}_m^{(n)}\right)
\end{aligned}$$

That is, we slice the matrices $\mathbf{G}_i$'s column wise and stitch them together by the column number in the original matrices. So $\widetilde{\mathbf{G}_i}$ is the $n \times m$ matrix composed of the ith columns of each of the $\mathbf{G}_1, \ldots, \mathbf{G}_m$ matrices.

For the Rectangular MinRank attack, we apply this deformation to the public matrices $(\mathbf{P}_1, \ldots, \mathbf{P}_m)$. We can see that since $\mathbf{P}_i = \mathbf{U}\mathbf{F}_i\mathbf{U}^\top$, we get $(\widetilde{\mathbf{P}}_1, \ldots, \widetilde{\mathbf{P}}_n) = (\mathbf{U} \cdot \widetilde{\mathbf{F}}_1, \ldots, \mathbf{U} \cdot \widetilde{\mathbf{F}}_n) \cdot \mathbf{U}^\top$. Therefore, if the $\widetilde{\mathbf{F}}_i's$ have some low rank property, then a linear combination of the $\widetilde{\mathbf{P}}_i$ should also be low rank.

A MinRank instance from the deformed matrices We now describe the Rectangular MinRank Attack applied to a Rainbow instance with L many layers.

We define $\mathcal{O}_L = \left\{ (\overbrace{0, \ldots, 0}^{n-o_L}, \overbrace{*, \ldots, *}^{o_L}) \in \mathbb{F}_q^n \right\}$ to be the Lth oil layer space in the private basis and $\mathbf{O}_L := \mathcal{O}_L\mathbf{U}^{-1}$ denote the Lth oil layer space in the public basis. The Lth oil layer space in the private basis $\mathcal{O}_L$ is the subspace of $\mathbb{F}_q^n$ where all of the central polynomials in $\mathcal{F}$ are guaranteed to vanish. Our goal is to find a non-zero element of $\mathbf{O}_L$. Since $\mathbf{U}$ is invertible, we know that $\mathbf{O}_L$ has dimension o_L.

Consider the subspace $A := \left\{ (\overbrace{*, \ldots, *}^{n-o_L+1}, \overbrace{0, \ldots, 0}^{o_L-1}) \in \mathbb{F}_q^n \right\}$ with $\dim(A) = n - o_L + 1$.

Since $\dim O_L = o_L$, Grassmann's inequality gives $\dim(O_L \cap A) \geq \dim O_L + \dim A - n = o_L + (n - o_L + 1) - n = 1$. This means that there is an element of $\mathbf{O}_L$ of the form $\mathbf{a} = (a_1, \ldots, a_{n-o_L+1}, 0, \ldots, 0)$.

Also notice that

$$\begin{aligned}
\sum_{i=1}^{n-o_L+1} a_i\widetilde{\mathbf{P}}_i &= \sum_{i=1}^{n} a_i\widetilde{\mathbf{P}}_i \qquad \text{(because the last } o_L \text{ entries are zero)} \\
&= (\widetilde{\mathbf{P}}_1, \ldots, \widetilde{\mathbf{P}}_n) \cdot \mathbf{a}^\top \\
&= (\mathbf{U}\widetilde{\mathbf{F}}_1, \ldots, \mathbf{U}\widetilde{\mathbf{F}}_n) \cdot (\mathbf{a}\mathbf{U})^\top.
\end{aligned}$$

And since $\mathbf{aU} \in \mathcal{O}_L$, its first $n - o_L$ coordinates are zero. This means, $\sum_{i=1}^{n-o_L+1} a_i \widetilde{\mathbf{P}}_i$ is a linear combination of $\mathbf{U}\widetilde{\mathbf{F}}_{n-o_L+1}, \ldots, \mathbf{U}\widetilde{\mathbf{F}}_n$. Since $\widetilde{\mathbf{F}}_{n-o_L+1}, \ldots, \widetilde{\mathbf{F}}_n$ have the following form,

$$\widetilde{F}_i = \begin{pmatrix} 0_{(n-o_L)\times(n-o_L)} & *_{(n-o_L)\times o_L} \\ 0_{(o_L)\times(n-o_L)} & 0_{(o_L)\times(o_L)} \end{pmatrix}$$

the subspace they span in $\mathbb{F}_q^{n\times m}$ is of rank at most $\min\{n-o_L, o_L\} = o_L$. When we multiply each of them by the full rank matrix $\mathbf{U}$, the resulting matrices form a subspace of $\mathbb{F}_q^{n\times m}$ of the same rank. Therefore, any of their linear combination is of rank at most o_L.

The rectangular MinRank attack on Rainbow aims to find a vector $\mathbf{a} \in \mathbf{O}_L$ by finding a solution to the MinRank problem

$$\text{Rank}\left(\sum_{i=1}^{n-o_L+1} a_i \widetilde{\mathbf{P}}_i\right) \leq o_L. \tag{3}$$

This MinRank instance has $n - o_L + 1$ many matrices of size $n \times m$ with target rank o_L.

5.2 Simple Attack

The Simple Attack of [6] landed the final blow that knocked Rainbow out of NIST standardization consideration. The Simple Attack utilizes the algebraic structure of Rainbow's polar forms.

As seen in Section 3.2, an instance of Rainbow relies on q, the characteristic of the finite field, n, the number of variables, m, the number of public key equations, and o_2, which is the dimension of both the second oil layer subspace $\mathbf{O}_2 \subset \mathbb{F}_q^n$ and the image of $\mathbf{O}_1$ under the public key $\mathcal{P}$. We will be consistent with the notation from [6] and set $\mathcal{P}(\mathbf{O}_1) = W \subset \mathbb{F}_q^m$.

$$\begin{array}{ccccc} \mathbf{O}_2 & \subset & \mathbf{O}_1 & \subset & \mathbb{F}_q^n \\ \downarrow\mathcal{P} & \searrow^{D\mathcal{P}(x,\cdot)} & \downarrow\mathcal{P} & & \downarrow\mathcal{P} \\ \{0\} & \subset & W & \subset & \mathbb{F}_q^m \end{array}$$

Fig. 3. Structure of nested subspaces.

From [6], it is known that for any $\mathbf{x} \in \mathbb{F}_q^n$ and any $\mathbf{o}_2 \in \mathbf{O}_2$ that $D\mathcal{P}(\mathbf{x}, \mathbf{o}_2) \in W$ and that $\mathcal{P}(\mathbf{O}_2) = \{0\}$. So, for a randomly chosen nonzero $\mathbf{x} \in \mathbb{F}_q^n$, construct the differential map

$$D_{\mathbf{x}} : \mathbb{F}_q^n \to \mathbb{F}_q^m, \mathbf{y} \mapsto D\mathcal{P}(\mathbf{x}, \mathbf{y}).$$

It is seen that $D_{\mathbf{x}}$ is a linear map which sends $\mathbf{O}_2$ to W. Given that $\dim(\mathbf{O}_2) = o_2 = \dim(W)$, the probability that $D_{\mathbf{x}}$ has a kernel vector in $\mathbf{O}_2$ is the same probability that a random $o_2 \times o_2$ matrix over $\mathbb{F}_q$ is singular. This probability is known to be

$$1 - \prod_{i=0}^{o_2-1} (1 - q^{i-o_2}) \approx q^{-1}.$$

In constructing this mapping, the goal is to find a nontrivial intersection between the kernel of the $D_{\mathbf{x}}$ map and the $\mathbf{O}_2$ subspace. To search for the possible nontrivial intersection, the Simple Attack sets up the following system of equations:

$$\begin{cases} D_{\mathbf{x}}(\mathbf{o}) = 0 \\ \mathcal{P}(\mathbf{o}) = 0. \end{cases} \tag{4}$$

The resulting system has m homogeneous linear equations and m homogenous quadratic equations in the n variables of $\mathbf{o}$. The Simple Attack then uses the m linear equations to eliminate m of the variables from the quadratic equations. This leaves a system of m homogenous equations in only $n - m$ variables. Using Beullens' notation, let $\mathbf{B} \in \mathbb{F}_q^{n \times (n-m)}$ be a matrix whose columns form a basis for $\ker(D_{\mathbf{x}})$. Then the process reduces to finding a solution to $\mathbf{o} \in \mathbb{F}_q^{n-m}$ such that $\mathcal{P}(\mathbf{Bo}) = 0$. If such a solution is found, then with high probability $\mathbf{o} \in \mathbf{O}_2$. If no solution exists, one randomly chooses a new nonzero $\mathbf{x} \in \mathbb{F}_q^n$ and repeat the process.

Once an $\mathbf{O}_2$ vector is found, the second layer of Rainbow can be removed and the complexity is reduced to an instance of UOV with $m - o_2$ equations in $n - o_2$ variables.

5.3 Combination of the Simple and Rectangular MinRank Attack

It was shown in [6] that a smaller MinRank instance can be obtained by combining the Simple Attack with the Rectangular MinRank attack. The goal of the combined attack remains the same: find an initial vector $\mathbf{a} \in \mathbf{O}_2$. However, instead of fixing the last $o_L - 1$ entries of the vector $\mathbf{a}$ to be 0 (as in the case of the original Rectangular MinRank), we reduce the number of unknown entries by using the fact that when choosing a random $\mathbf{x}$ we know that $D_{\mathbf{x}}(\mathbf{a}) = 0$ with probability q^{-1}.

We recall that for fixed $\mathbf{x} \in \mathbb{F}_q^n$, $D_{\mathbf{x}}$ is a linear map. Thus, using the m equations $D_{\mathbf{x}}(\mathbf{a}) = 0$ allows us to write m variables of $\mathbf{a}$ in terms of the $n - m$ remaining variables. This will allow us to rewrite the MinRank equation described in (3) into an equivalent MinRank equation with $n - m$ matrices.

As with the Simple Attack, the first step of the combined attack is to guess $\mathbf{x} \in \mathbb{F}_q^n$. The next step is to compute a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_{n-m}\}$ of $\ker(D_{\mathbf{x}})$. Next, an adversary will generate the matrices

$$\widetilde{L}_i := \begin{pmatrix} D\mathcal{P}(\mathbf{e}_1, \mathbf{b}_i) \\ \vdots \\ D\mathcal{P}(\mathbf{e}_n, \mathbf{b}_i) \end{pmatrix}, \quad 1 \leq i \leq n - m.$$

It then can be shown that $\mathbf{a} = \sum_{i=1}^{n-m} \alpha_i \mathbf{b}_i \in \ker D_{\mathbf{x}}$ is a solution of the original MinRank if and only if the vector $\alpha = (\alpha_1, \ldots, \alpha_n)$ is a solution to the new MinRank instance

$$\text{Rank} \left(\sum_{i=1}^{n-m} \alpha_i \widetilde{\mathbf{L}}_i \right) \leq o_L.$$

Since $D_{\mathbf{x}}(\mathbf{a}) = 0$ with a probability of $1/q$, we expect to restart the process q times to find an appropriate $\mathbf{x}$ value. Therefore, to get the complexity of finding the first vector, we need to multiply q by the complexity of solving the new MinRank instance. We will remove one row from each matrix to mitigate the syzygies, just as in the original Rectangular MinRank.

This gives us a MinRank instance with $n - m$ matrices of size $(n-1) \times m$ with target rank o_L. The complexity of this attack can be estimated as

$$\begin{aligned} \text{Exp Comp} &= (\text{Exp \# of guesses})(\text{Complexity of solving each MinRank instance}) \\ &= q \times (\text{Complexity given in Equation 1}). \end{aligned}$$

5.4 Padded rectangular MinRank attack

The padded MinRank attack was introduced in [16] in order to break the new parameter sets of VOX that were produced to counter attacks faced by the Rectangular MinRank attack in [12]. The VOX cryptosystem, introduced in [25], is a UOV-like cryptosystem which adds a small number t of purely random polynomials to a UOV central map. VOX was a Round One candidate in the Additional Digital Signature Schemes standardization for NIST [3], but did not make it into Round Two of the NIST competition.

An instance of VOX contains V vinegar variables, O oil variables, t random quadratic equations, and a total of o equations. In the new parameters of VOX, each of the deformed $\widetilde{\mathbf{F}}_i$ are now of full rank–providing security against the Rectangular MinRank attack. However, the authors of [16] note that the vertical concatenation of λ many central maps will have a rank anomaly. More specifically, the vertical concatenation of $\widetilde{\mathbf{F}}_{N-1}$ with $\widetilde{\mathbf{F}}_N$ yields a matrix of rank at most $2V + t$. This is due to the fact that $2O > t$ and $m - t > 2V$ for the modified parameters. This is illustrated in Figure 4.

Generally, for $\lambda \leq O$, if $m - t > \lambda v$ and $\lambda O > t$, then the matrix

$$\mathbf{M}'_s = \begin{bmatrix} \mathbf{U}\widetilde{\mathbf{F}}_{N-\lambda+1}\mathbf{T} \\ \vdots \\ \mathbf{U}\widetilde{\mathbf{F}}_N\mathbf{T} \end{bmatrix} = (\mathbf{I}_\lambda \otimes \mathbf{U}) \begin{bmatrix} \widetilde{\mathbf{F}}_{N-\lambda+1} \\ \vdots \\ \widetilde{\mathbf{F}}_N \end{bmatrix} \mathbf{T}$$

has rank at most $\lambda V + t$.

We will solve an instance of MinRank to find column vectors of $(\mathbf{U}^\top)^{-1}$, which will allow us to find λ many vectors in $\mathcal{O}_2\mathbf{U}^{-1}$. Given that $(\widetilde{\mathbf{P}}_1, \ldots, \widetilde{\mathbf{P}}_n) = (\mathbf{U} \cdot \widetilde{\mathbf{F}}_1 \cdot \mathbf{T}, \ldots, \mathbf{U} \cdot \widetilde{\mathbf{F}}_n \cdot \mathbf{T}) \cdot \mathbf{U}^\top$, we can rewrite $\mathbf{M}'_s$ as

[3] https://csrc.nist.gov/Projects/pqc-dig-sig/round-2-additional-signatures

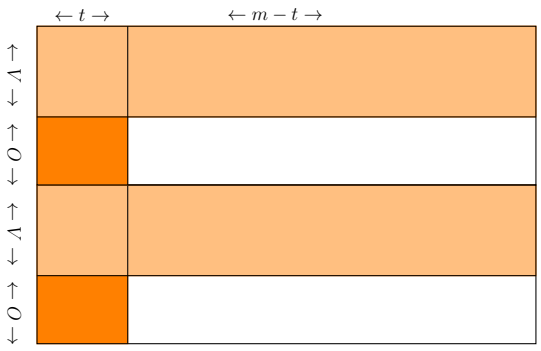


Fig. 4. The shape of $\begin{bmatrix} \widetilde{\mathbf{F}}_{N-1} \\ \widetilde{\mathbf{F}}_N \end{bmatrix}$ for VOX. The rank does not exceed $2V + t$.

$$\mathbf{M}'_s = \begin{bmatrix} \sum_{i=1}^{N} x_{1,i}\widetilde{\mathbf{P}}_i \\ \vdots \\ \sum_{i=1}^{N} x_{\lambda,i}\widetilde{\mathbf{P}}_i \end{bmatrix},$$

where $\mathbf{x}_j = (x_{j,1}, \ldots, x_{j,N})$ is the $N-\lambda+j$ column of $(\mathbf{U}^\top)^{-1}$. In VOX, $(\mathbf{U}^\top)^{-1}$ is a block upper triangular matrix and therefore have $x_{j,i} = \delta_{i,N-\lambda+j}$ for $i > V$.

Then we can define $\mathbf{M}_s$ as

$$\mathbf{M}_s = \begin{bmatrix} \sum_{i=1}^{V} x_{1,i}\widetilde{\mathbf{P}}_i + \widetilde{\mathbf{P}}_{N-\lambda+1} \\ \vdots \\ \sum_{i=1}^{V} x_{\lambda,i}\widetilde{\mathbf{P}}_i + \widetilde{\mathbf{P}}_N \end{bmatrix}.$$

Solving for the $\mathbf{x}_j$'s gives us an inhomogeneous MinRank instance. So, each component of the linear combination has the form

$$\widetilde{\mathbf{F}}_i^{(j,\lambda)} = \begin{bmatrix} 0 \cdots & \widetilde{\mathbf{F}}_i & \cdots 0 \end{bmatrix}^\top,$$

where λ is the number of matrices concatenated.

Section 3.1 of [16] shows that there are nonzero linear combinations of the first $2N$ rows of $\mathbf{M}_s$ due to the symmetry present in the central and public maps. Syzygies exist for every pair of blocks, which means $\mathbf{M}_s$ will have rank at most $\lambda N - \binom{\lambda}{2}$.

We can also consider this attack to try to find a vector in the outermost layer of a Rainbow cryptosystem. Since Rainbow does not use random polynomials, $t = 0 < o_L(o_L + 1)/2$. We see that this attack elicits a smaller target rank when applied against Rainbow. When we stack two or more deformed central matrices from $\widetilde{\mathbf{F}}_{n-o_L+1}, \ldots, \widetilde{\mathbf{F}}_n$, the maximum rank we can get is o_L because $o_L < \lambda N$. We illustrate two stacked central map matrices for Rainbow in Figure 5.

Although not all linear maps used for Rainbow can be written in terms of block triangular matrices, we can define the same $\mathbf{M}_s$ using a similar argument to the one used in Equation (3). Namely, there are elements $\mathbf{x}_j$s, $1 \leq j \leq \lambda$ of $\mathbf{O}_L$ of the form $\mathbf{x}_j = (x_{j,1}, \ldots, x_{j,N-O_L}, 0, \ldots, x_{j,N-\lambda+j}, 0, \ldots)$, that is, the last O_L coordinates are 0 except at $N - \lambda + j$.

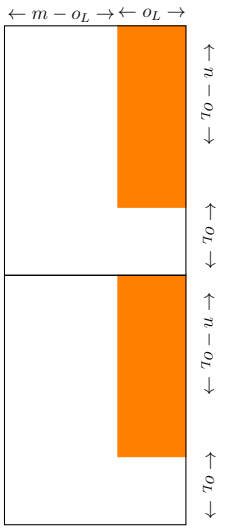


Fig. 5. Stacking two Rainbow deformed central map matrices $\tilde{\mathbf{F}}_i$ and $\tilde{\mathbf{F}}_j$, where $n - o_L + 1 \leq i, j \leq n$.

The Padded Rectangular MinRank attack on Rainbow aims to find solutions $\mathbf{x}_j \in \mathbf{O}_L$ such that

$$\text{Rank}\left(\begin{bmatrix} \sum_{i=1}^{N-o_L} x_{1,i}\widetilde{\mathbf{P}}_i + \widetilde{\mathbf{P}}_{N-\lambda+1} \\ \vdots \\ \sum_{i=1}^{N-o_L} x_{\lambda,i}\widetilde{\mathbf{P}}_i + \widetilde{\mathbf{P}}_N \end{bmatrix}\right) \leq o_L. \tag{5}$$

This results in a MinRank instance with $\lambda(N - o_L)$ many $\lambda n \times m$ matrices with target rank o_L.

5.5 Combination of the Simple and the Padded MinRank Attack

We introduce a new cryptanalytic technique that combines the Simple Attack with the Padded MinRank Attack. Because the Simple Attack relies on the existence of intermediate Rainbow subspaces, it does not apply to the VOX cryptosystem and therefore does not improve upon the cryptanalysis of VOX presented in [16]. While this combination is not applicable to every instantiation

of the Padded MinRank Attack, it applies to Rainbow-like schemes with a public map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and hidden spaces $O \subseteq \mathbb{F}_q^n$, $W \subseteq \mathbb{F}_q^m$ satisfying $\mathcal{P}(O) = 0$, $D\mathcal{P}(\mathbf{x}, O) \subseteq W$ for every $\mathbf{x}$, $\dim O = \dim W$, and such that the deformed matrices associated with vectors in O, including the vertically stacked matrices used by the attack, have rank at most the target rank r. The first three conditions enable the Simple Attack, while the last preserves the padded target rank. VDOO satisfies these conditions for $O = \mathbf{O}_2$ and $W = \mathcal{P}(\mathbf{O}_1)$, as established in Section 5.4.

The goal of the Padded MinRank attack is to find λ many linearly independent vectors of $\mathbf{O}_L$. To combine it with the Simple Attack, we first randomly sample a vector $\mathbf{x} \in \mathbb{F}_q^n$. We hope to find an $\mathbf{x}$ such that the kernel of $D_{\mathbf{x}}$ contains a vector in $\mathbf{O}_L$. But in the case of Rainbow, we are not guaranteed any specific number of linearly independent vectors in the intersection of $D_{\mathbf{x}}$ and $\mathbf{O}_L$. We also cannot force more constraints on the coefficients because the dimension of $D_{\mathbf{x}}$ is $n - m$ and the dimension of $\mathbf{O}_L$ is o_L, which does not allow us to make any assumptions about their intersection. If we impose too many restrictions, we risk not finding any solutions.

One viable way to still combine the two attacks is to sample λ many vectors, $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$, one for each stacked matrix. As with the case of the original Padded MinRank Attack, the target rank of the MinRank instance will be o_L as we are stacking the matrices vertically. With each iteration, we may find up to λ many vectors in $\mathbf{O}_L$, but there is no guarantee of linear independence. This is not an issue, though, as we can follow the steps of Section 7 to find more vectors in $\mathbf{O}_L$.

Completing this attack is mathematically equivalent to running λ many instances of the combined Simple and Rectangular MinRank attack independently in parallel. The goal is to find at least one $\mathbf{x}_i \in \{\mathbf{x}_1, \ldots, \mathbf{x}_\lambda\}$ that gives the desired results. Therefore, we can expect to run the process about q/λ times to get one vector in $\mathbf{O}_L$. The Combined Simple and Padded MinRank Attack leads to a MinRank instance containing λv many matrices that contain $\lambda(n-1)$ rows, m columns and with target rank of o_L, that we expect to be repeated around q/λ times. Therefore, the complexity can be estimated as

$$\begin{aligned}\text{Exp Comp} &= (\text{Exp \# of guesses})(\text{Complexity of solving each MinRank instance}) \\ &\approx \left(\frac{q}{\lambda}\right) \times (\text{Complexity given in Equation 1}).\end{aligned}$$

6 Attacking VDOO: Finding a Layer 2 Oil Vector

We discuss the updated cryptanalysis of VDOO, focusing on the attacks described in Sections 5.3 and 5.5. The original VDOO paper considers the attacks described in Section 5.3, but concludes that the security is higher than closer examination suggests. The first and dominating step when using either the Rectangular MinRank attack or the Padded MinRank attack is to solve the MinRank problems given in Equations 3 and 5. This step outputs a non-zero vector in the

secret $\mathbf{O}_2$ space. Once this vector is obtained, we follow the steps in Section 7 to determine each UOV layer.

For easy reference, the parameters of VDOO and its claimed security are presented in Table 1.

Sec Level	Claimed Sec	q	v	d	o_1	o_2
SL-I	134	16	60	30	34	36
SL-III	191	256	100	30	40	40
SL-V	264	256	120	50	60	70

Table 1. VDOO parameters. The Claimed Security (in bits) of VDOO parameter sets, as given in Table 1 of [14].

6.1 Heuristics and Assumptions for the Complexity Estimates.

Our security estimates combine exact structural reductions with heuristic algebraic complexity models presented in [6] and other work. The dimensions and target ranks of the MinRank instances, as well as the preservation of valid oil-space solutions under row deletion, are deterministic. The estimated cost of solving the resulting MinRank instances uses the Support Minors dimension count. We assume that the corresponding Macaulay matrices attain the predicted rank at the reported degree b. For $\lambda = 1$, deleting one row from each matrix is used to avoid the characteristic-two syzygies discussed in [6]. For $\lambda > 1$, cross-block syzygies may remain, so the resulting Support Minors estimates should be viewed as heuristic projections rather than validated average-case estimates.

For the Simple Attack, we model the restriction $D_x|_{O_2} : O_2 \to W$ as a uniformly random linear map between spaces of equal dimension. Its probability of being singular is

$$p_{q,o_2} = 1 - \prod_{j=1}^{o_2}(1 - q^{-j}), \tag{6}$$

so the expected number of independent trials is $1/p_{q,o_2}$. When λ independent samples are used, the corresponding success probability is $1 - (1 - p_{q,o_2})^{\lambda}$. In the tables, we use the first-order approximation q/λ, which is appropriate when λ is small relative to q.

Finally, the basis-completion estimates in this section assume that the resulting quadratic systems exhibit semi-regular behavior through the stated witness degree. Equation (1) counts field operations; we multiply this estimate by the trial factor used above and report the base-two logarithm of the resulting cost. These estimates do not include a separate memory-access penalty or a full-parameter wall-clock measurement.

6.2 Simple Attack and Rectangular MinRank Applied to VDOO

Here we provide a thorough and accurate complexity analysis of the combined Simple and Rectangular MinRank Attack on VDOO, as well as code enabling experimental validation. The original VDOO paper [14] presents a security estimate against these attacks, but arrives at an overly optimistic complexity assessment.

The structure of the VDOO central maps is illustrated in Figure 2. If we were implementing the Rectangular MinRank Attack on its own, we would consider the matrix representation of the deformation of the public equations of VDOO. Instead, we consider the Combined Simple and Rectangular MinRank Attack.

We start by initializing the Simple Attack. Given a VDOO public key, we generate a random vector $\mathbf{x}$ and find a basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_{n-m}\}$ of $\ker(D_{\mathbf{x}})$. We then construct the $\widetilde{L}_i$ matrices, and solve the MinRank instance

$$\text{Rank}\left(\sum_{i=1}^{n-m} \alpha_i \widetilde{\mathbf{L}}_i\right) \leq o_L. \tag{7}$$

It was shown in [6] that, over fields of characteristic 2, the Support Minors polynomials obtained from this system contain syzygies that are absent from random MinRank instances. These syzygies make the resulting polynomial system harder to solve than a random instance with the same degree and number of variables. If these syzygies are not accounted for, the complexity of solving a random MinRank instance provides only a lower bound for the complexity of solving the specific instance, rather than an estimate of its average complexity. To avoid these syzygies, we delete one row from each matrix and apply the complexity estimate from [2].

Deleting a row from each matrix cannot increase its rank. Thus, every coefficient vector that yields rank at most o_2 in the original MinRank instance also satisfies the truncated instance. The converse need not hold, however, so truncation may introduce spurious candidates. We therefore verify all candidates obtained from the truncated system against the original matrices. Experiments reported in Appendix A of [6] indicate that the expected number of such spurious candidates is small for Rainbow instances. Accordingly, we use the reduced row count in our complexity estimates, while acknowledging that this approach assumes similar behavior for the VDOO instances considered here.

We let $\widetilde{\mathbf{Q}}_i$ denote the matrix obtained by removing the last row of $\widetilde{\mathbf{L}}_i$. The system then becomes

$$\text{Rank}\left(\sum_{i=1}^{v} \alpha_i \widetilde{\mathbf{Q}}_i\right) \leq o_2. \tag{8}$$

Therefore, the Combination Simple + Rectangular MinRank attack against VDOO consists of a MinRank instance containing v many matrices of size $(n-1) \times m$ with target rank o_2.

We provide the estimated complexity of finding a layer 2 oil vector using the Rectangular MinRank attack in Table 3 for each of the provided parameter

sets. We note that the analysis provided here gives a lower estimated complexity than that provided in [14]. The overestimate in [14] appears to result from evaluating the Support-Minors complexity on the full column MinRank instance, rather than minimizing over admissible column truncations; using the optimal 47-column truncation, as opposed to the full 100 columns, reduces the SL-I estimate from approximately 2^{133} to 2^{106} field operations.

We note that it was shown in [6], that the equations $P_1(\mathbf{a}) = \cdots = P_m(\mathbf{a}) = 0$ can be added to the equations of the MinRank system. We did not include this step into our analysis as the combination of the simple attack and the rectangular MinRank attack is enough to break all the parameters proposed in the VDOO paper [14].

We provide Magma code that runs the first step of the attack on VDOO instances in this GitHub repository. Given an instance of a VDOO public key, the code will return a vector in o_2. Note that our code uses the Kipnis-Shamir Modeling of MinRank, which is not guaranteed to be the most effective for all parameters. Some experimental results are reported in Appendix A.

	Rectangular MinRank instance parameters			
VDOO Param Set	M(#rows)	N(#cols)	K(#mat)	r
(q, v, d, o_1, o_2)	$v + d + o_1 + o_2 - 1$	$d + o_1 + o_2$	v	o_2
SL-I (16,60,30,34,36)	159	100	60	36
SL-III (256,100,30,40,40)	209	110	100	40
SL-V (256,120,50,60,70)	299	180	120	70

Table 2. Parameters of MinRank instances in the Simple Attack combined with Rectangular MinRank. We expect to complete q many MinRank computations to get one vector in $\mathbf{O}_2$.

Combination Simple Attack and Rectangular MinRank Attack				
Claimed Sec	VDOO params	m'	b	Bit Operations
134	SL-I	47	2	106
191	SL-III	53	3	136
264	SL-V	91	3	194

Table 3. Complexity of the Simple Attack combined with Rectangular MinRank using Support Minors modeling on VDOO parameters from Table 2. The columns b (m') is the degree (resp. number of columns) that gives the minimal complexity.

6.3 Simple Attack and Padded MinRank Applied to VDOO

We now apply the novel Combined Simple and Padded MinRank Attack to the provided parameters of VDOO, keeping the same notation as the previous

section. Note that the Padded MinRank Attack with $\lambda = 1$ is the same as the Combined Simple and Rectangular MinRank (where λ represents the number of stacked central maps considered). It was shown in [6] that in the case of $\lambda = 1$, it is possible to side-step the syzygies by removing one row of each matrix. In the case $\lambda > 1$, not only do we have the syzygies from each block, we also have syzygies for every pair of blocks.

For $\lambda > 1$, deleting one row from each block removes the characteristic-two syzygies internal to each block but does not eliminate syzygies involving multiple blocks. We therefore treat the resulting $\lambda > 1$ complexity estimates as heuristic projections rather than rigorous average-case estimates, and verify all candidates against the original stacked matrices.

We first randomly generate $\mathbf{x}_1, \ldots, \mathbf{x}_\lambda$. We then construct their respective Simple and Rectangular MinRank instances. For $\mathbf{x}_i$, $1 \leq i \leq \lambda$, the matrices will be denoted $\widetilde{\mathbf{Q}}_{i,j}$, for $1 \leq j \leq v$.

We then consider solving the MinRank instance

$$\text{Rank}\left(\begin{bmatrix} \sum_{j=1}^{v} x_{1,j}\widetilde{\mathbf{Q}}_{1,j} \\ \vdots \\ \sum_{j=1}^{v} x_{\lambda,j}\widetilde{\mathbf{Q}}_{\lambda,j} \end{bmatrix}\right) \leq o_2. \tag{9}$$

This can be written as

$$\text{Rank}\left(\sum_{j=1}^{v} x_{1,j}\begin{bmatrix} \widetilde{\mathbf{Q}}_{1,j} \\ \vdots \\ 0 \end{bmatrix} + \cdots + \sum_{j=1}^{v} x_{\lambda,j}\begin{bmatrix} 0 \\ \vdots \\ \widetilde{\mathbf{Q}}_{\lambda,j} \end{bmatrix}\right) \leq o_2. \tag{10}$$

This is a MinRank instance with λv matrices of size $(\lambda(n - o_2 - 1) \times m)$. We expect the complexity to be similar to the complexity of a random MinRank instance with λv matrices of size $(\lambda(n-1) \times m)$.

Using Equations 1 and 2, we provide the complexity estimates of obtaining a vector in $\mathbf{O}_2$ using the Padded MinRank attack against VDOO in Table 5. Once again, the steps to follow once this vector is obtained are outlined in Section 7. We can see that, at least for these parameters, the Rectangular MinRank attack is expected to outperform the Padded Rectangular attack with more layers. We provide the MinRank instance parameters obtained at this step in Table 4.

7 Completing the Attacks

Once a vector $\mathbf{a} \in \mathbf{O}_2$ is found, an adversary is able to uncover all of the secret spaces of the VDOO cryptosystem. Here we summarize the necessary steps to achieve this and provide concrete security estimates for carrying out these steps against VDOO. We conclude that, despite VDOO containing many secret kernel spaces across multiple layers, this structure does not provide sufficient protection against an adversary.

	MinRank instance parameters			
VDOO Param Set	M(#rows)	N(#cols)	K(#var)	r
(q, v, d, o_1, o_2)	$(v + d + o_1 + o_2 - 1)\lambda$	$d + o_1 + o_2$	λv	o_2
SL-I (16,60,30,34,36)	159 λ	100	60 λ	36
SL-III (256,100,30,40,40)	209 λ	110	100λ	40
SL-V (256,120,50,60,70)	299 λ	180	120 λ	70

Table 4. MinRank instances obtained by applying the Padded MinRank Attack with λ many stacked matrices to VDOO. We expect to complete approximately q/λ many MinRank computations to get one vector in $\mathbf{O}_2$.

Combination Simple Attack and Padded MinRank Attack										
		$\lambda = 1$			$\lambda = 2$			$\lambda = 3$		
Claimed Sec	VDOO params	m'	b	Bit Comp	m'	b	Bit Comp	m'	b	Bit Comp
134	SL-I	47	2	106	50	1	110	50	1	111
191	SL-III	53	3	136	56	2	141	60	1	142
264	SL-V	91	3	194	90	3	195	90	3	199

Table 5. Bit complexity of completing the combination Simple Attack and Padded MinRank Attack Using Support Minors. We note that in the case of $\lambda = 1$, the Padded MinRank attack is equivalent to the Rectangular MinRank attack.

We can describe the spaces used in VDOO as a system of nested subspaces, similar to that of Rainbow and described in [5]. The nested property of the subspaces allows one to use the public key and a vector from the Layer 2 Oil space to obtain an equivalent key.

Recall, we let $\mathbf{O}_i$ denote the ith Oil Layer and extend this notation to let $\mathbf{D}_j$ denote the jth Diagonal Layer space in the public basis. I.e., $\mathcal{O}_i\mathbf{U}^{-1} := \mathbf{O}_i$ and $\mathcal{D}_j\mathbf{U}^{-1} := \mathbf{D}_j$. We let $\mathbf{Y}_j$ denote the image of the jth layer of the Diagonal space under the public equations $\mathcal{P}$, and $\mathbf{W} := \mathcal{P}(\mathbf{O}_1)$. Notice that $\mathcal{P}$ vanishes on the subspace $\mathbf{O}_2$, and that $\dim(\mathbf{O}_2) = o_2$. Figure 6 illustrates the nested properties of the involved spaces.

In order to forge a signature, we will apply the Rectangular MinRank attack multiple times to reveal a nonzero element from each hidden subspace. During this process, we will also uncover $\mathbf{U}^{-1}$, which will allow us to solve for the central map $\mathcal{F}$.

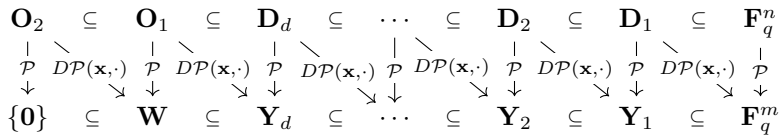


Fig. 6. The nested subspaces involved in the VDOO cryptosystem. Note that $\mathbf{x} \in \mathbb{F}_q^n$, $\mathcal{P}$ denotes the public equations, and $D\mathcal{P}$ denotes the polar form.

Both the Padded MinRank and Rectangular MinRank attacks return vectors in $\mathbf{O}_2$. The Rectangular MinRank Attack will return one vector from $\mathbf{O}_2$, whereas the Padded MinRank Attack will return λ many $\mathbf{O}_2$ vectors. Once a vector $\mathbf{a} \in \mathbf{O}_2$ is found, we use this to complete a basis of $\mathbf{O}_2$. If $\lambda < o_2$ in the Padded MinRank attack, we complete a basis of $\mathbf{O}_2$ in the same manner as completing a basis for $\mathbf{O}_2$ in the Rectangular MinRank attack.

Given a vector $\mathbf{a}_1 \in \mathbf{O}_2$, we can find a second vector $\mathbf{a}_2 \in \mathbf{O}_2$ by solving

$$\begin{cases} \mathcal{P}(\mathbf{a}_2) = 0 \\ D\mathcal{P}(\mathbf{a}_1, \mathbf{a}_2) = 0, \end{cases}$$

as described in Section 3.3 of [6].

Note that $D\mathcal{P}(\mathbf{a}_1, \mathbf{a}_2) = 0$ is linear in $\mathbf{a}_2$ because $D\mathcal{P}$ is bilinear. So this system is made of m quadratic equations and m linear equations in n variables, which represent the entries of $\mathbf{a}_2$. Using the linear equations, we can write m entries of $\mathbf{a}_2$ as functions of the remaining entries using Gaussian elimination. The cost of Gaussian elimination is $2m^3/3$ arithmetic operations.

After that, we need to solve m quadratic equations with $n - m$ unknowns. Since this system is overdetermined, we can solve it using the XL algorithm. Assuming the equations behave like random systems, the cost will be

$$3\binom{n - m + d_{wit}}{d_{wit}}\binom{n - m}{2},$$

where d_{wit} is as defined in Definition 2 of [3], and is an estimate for the highest degree reached in the XL algorithm.

When we have a second vector, we can iteratively find the remaining $o_2 - 2$ vectors to complete the basis by solving

$$\begin{cases} \mathcal{P}(\mathbf{a}_k) = 0 \\ D\mathcal{P}(\mathbf{a}_i, \mathbf{a}_k) = 0 \quad \forall 1 \leq i < k. \end{cases}$$

The complexity of solving these equations decreases with i, so we are expecting at most

$$(o_2 - 1) \cdot \left(3\binom{n - m + d_{reg}}{d_{reg}}\binom{n - m}{2} + \frac{2m^3}{3}\right)$$

operations for the total process of finding $\mathbf{O}_2$ after $\mathbf{a}_1$ is found. Table 6 gives the complexity of finding all o_2 after finding the first vector

Once $\mathbf{O}_2$ is found, we can then compute $\mathbf{W} = \mathcal{P}(\mathbf{O}_1)$. This is because for each discovered $\mathbf{a}_k$, the rows of matrix $L_{\mathbf{a}_k} = \sum_{i=1}^{n} \mathbf{a}_{k_i} \widehat{P}_i$ are in $\mathbf{W}$.

Each $L_{\mathbf{a}_k}$ is obtained by computing $L_{\mathbf{a}_k} = \sum_{i=1}^{n} \mathbf{a}_k L_{\mathbf{e}_i}$ which requires $n^2 m$ multiplications. There are $o_2 n$ rows in total. We then need to find o_2 many linearly independent vectors of this form to obtain a basis of $\mathbf{W}$. This is done using Gaussian Elimination. The cost of Gaussian elimination is about $2(no_2)^3/3$ multiplications and is again negligible compared to the cost of finding $\mathbf{a}_1$.

Claimed Sec	VDOO params	find $\mathbf{a}_1$ via Simple	find basis $\mathbf{O}_2$	Total finding $\mathbf{O}_2$
134	SL-I	106	66	106
191	SL-III	136	134	136
264	SL-V	194	114	194

Table 6. The cost (in bits of operations) of finding remaining vectors in $\mathbf{O}_2$ after the first vector is found via Simple attack combined with Rectangular MinRank attack. The total complexity of finding $\mathbf{O}_2$ is the bit complexity of the sum of the two steps. We notice the complexity is dominated by finding the first vector in $\mathbf{O}_2$.

After $\mathbf{W}$ is found, we are left with a smaller rainbow instance from $\mathbb{F}_q^n/\mathbf{O}_2$ to $\mathbb{F}_q^m/\mathbf{W}$ with v vinegar variables, d layers of 1 oil variable each and the layer with o_1 variables. We give the parameters and attack costs of the smaller Rainbow layer in Table 7 and write it with notation consistent with the VDOO parameters.

Notice that the solving for the $\mathbf{O}_1$ subspace has slightly higher complexity than the solving for the $\mathbf{O}_2$ layer. However, the complexity of completing this attack is still below the security claim for all three proposed parameter sets.

Claimed Sec	VDOO params	Cost $\mathbf{O}_2$	Smaller VDOO instance	Cost smaller instance	Cost $\mathbf{O}_2$&$\mathbf{O}_1$
134	SL-I	106	(16,60,29,1,34)	116	116
191	SL-III	136	(256,100,29,1,40)	150	150
264	SL-V	194	(256,120,49,1,60)	196	196

Table 7. Parameters and complexity (in bits of operations) of completing the Combination Rectangular MinRank Attack. The column "Cost $\mathbf{O}_2$" gives the estimated cost of uncovering the outermost oil layer. Columns 4 and 5 give the parameters and estimated cost of performing the Rectangular MinRank attack against the smaller Rainbow instance from $\mathbb{F}_q^n/\mathbf{O}_2$ to $\mathbb{F}_q^m/\mathbf{W}$ after the $\mathbf{O}_2$ layer is removed. The column "Cost $\mathbf{O}_2$&$\mathbf{O}_1$" denotes the cost of removing the two layers combined.

Using the same process, we continue removing the layers one by one until we know the innermost oil layer. An adversary can then use this to sign a message that verifies under the public key. Table 8 gives the cost of uncovering the first five outermost diagonal of each VDOO parameter set.

Once all of the subspaces are recovered, the adversary has different options for how to proceed. First, they may reconstruct the matrix $\mathbf{U}^{-1}$ and compute the central map $\mathcal{F}$ using $\mathbf{F}_i = \mathbf{U}^{-1}\mathbf{P}_i(\mathbf{U}^{-1})^\top$, enabling signature generation with the standard algorithm. Alternatively, we can follow the procedure described in Section 5 of [5].

Claimed Sec	Params	$\mathbf{O}_2$	$\mathbf{O}_1$	$\mathbf{D}_d$	$\mathbf{D}_{d-1}$	$\mathbf{D}_{d-2}$	$\mathbf{D}_{d-3}$	$\mathbf{D}_{d-4}$	Total Est Cost
134	SL-I	106	116	27	27	27	27	27	116
191	SL-III	136	150	34	34	34	34	34	150
264	SL-V	194	196	34	34	34	34	34	196

Table 8. Cost (in bits of operations) of unwrapping the layers of VDOO using the Combination Rectangular MinRank method. The columns $\mathbf{O}_1$ and $\mathbf{O}_2$ give the cost of uncovering the oil layers. The columns titled $\mathbf{D}_i$, $1 \leq i \leq 4$ give the estimated cost of uncovering the ith diagonal layer. The total estimated cost is the base 2 logarithm of the sum of the complexities of each step.

8 General Cryptanalysis and Comparison to Rainbow

Here we show how the structural relationship between VDOO and Rainbow can be used to adapt attacks created for Rainbow to be used against VDOO. Throughout, parameters with a superscript (R) denote parameters as described in the context of Rainbow.

Recall that a Rainbow signature scheme has $L^{(R)}$ many layers, where each layer ℓ contains $o_\ell^{(R)}$ many oil variables and $v_\ell^{(R)}$ many vinegar variables, satisfying $o_\ell^{(R)} + v_\ell^{(R)} = v_{\ell+1}^{(R)}$ where $v_{L^{(R)}+1}^{(R)} = n^{(R)}$. Given an instance of VDOO with parameters (d, v, o_1, o_2), we can translate this into a Rainbow instance with $L^{(R)} = d + 2$ many layers and the following parameters:

$$v_i^{(R)} = \begin{cases} v + i - 1 & \text{for } 1 \leq i \leq d+1 \\ v + d + o_1 & \text{for } i = d+2, \end{cases} \qquad o_i^{(R)} = \begin{cases} 1 & \text{for } 1 \leq i \leq d \\ o_1 & \text{for } i = d+1 \\ o_2 & \text{for } i = d+2. \end{cases}$$

We can see from Table 9 that the most significant difference between the proposed parameters for VDOO and those of Rainbow in the third round of the NIST submission process is the number of layers, with VDOO using 32, 32, and 52, whereas Rainbow uses 2 throughout. The reasoning for the fixed layer choice in Rainbow is explained in Section 3.4 of the Rainbow 3rd Round Submission Packet [22], [7]:

> "The Rainbow signature scheme can be defined for any number of layers u. For $u = 1$ we obtain the well known UOV signature scheme [18]. However, choosing $u = 2$ leads to a scheme with more efficient computations and smaller key sizes at the same level of security. Choosing $u > 2$ gives only a very small benefit in terms of performance, but needs larger keys to reach the same security level. Therefore, for ease of implementation, 2 is a common choice for the number of Rainbow layers."

Comparing the two schemes, we observe a substantial increase in the number of layers and a modest increase in the total number of variables (VDOO parameters are approximately 1.6 times larger than the corresponding Rainbow

parameters). In contrast, the size of the outermost oil space—which corresponds to the target rank in the Simple Attack—remains comparable, with the parameters of VDOO being only about 1.1 times larger than those of Rainbow.

Rainbow Layer ℓ	VDOO Layer	# eq and $o_\ell^{(R)}$	$v_\ell^{(R)}$	dim of kernel
1	Diag	1	v	$o_1 + o_2 + d$
2	Diag	1	$v+1$	$o_1 + o_2 + d - 1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
d	Diag	1	$v + d - 1$	$o_1 + o_2 + 1$
$d+1$	Oil 1	o_1	$v + d$	$o_1 + o_2$
$d+2$	Oil 2	o_2	$v + d + o_1$	o_2

Table 9. This table illustrates how to translate a (v, d, o_1, o_2) VDOO scheme into the language of a Rainbow scheme.

		Rainbow parameters			VDOO parameters		
Param Set	q	$L^{(R)}$	$n^{(R)}$	$o^{(R)}_{L^{(R)}}$	$L^{(R)}$	$n^{(R)}$	$o^{(R)}_{L^{(R)}}$
I	16	2	100	32	32	160	36
III	256	2	136	36	32	210	40
V	256	2	196	64	52	300	70

Table 10. We summarize the parameters of Rainbow (3rd-round NIST submission, [7]) and VDOO for the proposed security levels I, III, and V, using the notation of a Rainbow cryptosystem. In particular, we highlight the total number of variables $n^{(R)}$, the number of layers $L^{(R)}$, and the size of the oil space in the outermost layer $o^{(R)}_{L^{(R)}}$.

9 Conclusion

Our analysis shows that the security of VDOO depends mostly on the complexity of recovering the first and second oil layer (two layers with the smallest secret kernel space), while the diagonal structure has a negligible effect on the security. This implies that VDOO falls under the umbrella of Rainbow schemes, instead of being a new branch in the UOV family tree. Consistent with this view, the proposed VDOO parameter sets fail to meet their claimed security levels, and the concrete attack complexities are comparable to those for Rainbow instances with similar parameters.

Multivariate cryptography continues to evolve through the introduction of new structural designs, which in turn motivate new forms of structural cryptanalysis. As such attacks emerge, it is essential to evaluate their impact across

a wide range of settings and to compare competing techniques under consistent assumptions in order to obtain reliable security estimates and identify the most effective cryptanalytic strategy.

In this work we provide an updated and systematic cryptanalysis of the VDOO cryptosystem. We showed that all provided parameter sets do not achieve their advertised security levels–not due to the intrinsic VDOO structure, but because the parameters of the outer Oil Layer remain vulnerable to known attacks.

As the deployment of post-quantum cryptography accelerates, thorough and accurate cryptanalysis is increasingly critical. This need is underscored by NIST's continued emphasis on schemes based on diverse hardness assumptions to ensure a robust public-key infrastructure. Our results contribute to this effort by clarifying the true security estimates of VDOO and by providing a more reliable basis for future parameter selection and design in Rainbow-like systems.

Statements and Declarations

Competing Interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Adj, G., Aragon, N., Barbero, S., Bardet, M., Bellini, E., Bidoux, L., Chi-Domínguez, J.J., Dyseryn, V., Esser, A., Feneuil, T., Gaborit, P., Neveu, R., Rivain, M., Rivera-Zamarripa, L., Sanna, C., Tillich, J.P., Verbel, J., Zweydinger, F.: Mirath signature scheme: Algorithm specifications and supporting documentation. Tech. Rep. Version 2.0, NIST / CSRC (May 2025), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/mirath-spec-round2-web.pdf, accessed: 2025-10-17
2. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R.A., Smith-Tone, D., Tillich, J., Verbel, J.A.: Improvements of algebraic attacks for solving the rank decoding and minrank problems. In: Moriai, S., Wang, H. (eds.) 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. Lecture Notes in Computer Science, vol. 12491, pp. 507–536. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_17, https://doi.org/10.1007/978-3-030-64837-4_17
3. Bardet, M., Faugère, J., Salvy, B., Spaenlehauer, P.: On the complexity of solving quadratic boolean systems. J. Complex. **29**(1), 53–75 (2013). https://doi.org/10.1016/J.JCO.2012.07.001, https://doi.org/10.1016/j.jco.2012.07.001
4. Bellini, E., Makarim, R.H., Sanna, C., Verbel, J.: An estimator for the hardness of the MQ problem. Lecture Notes in Computer Science (2022). https://doi.org/10.1007/978-3-031-17433-9_14
5. Beullens, W.: Improved cryptanalysis of UOV and rainbow. In: Canteaut, A., Standaert, F. (eds.) 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. Lecture Notes in Computer Science, vol. 12696, pp. 348–373. Springer (2021). https://doi.org/10.1007/978-3-030-77870-5_13, https://doi.org/10.1007/978-3-030-77870-5_13

6. Beullens, W.: Breaking rainbow takes a weekend on a laptop. IACR Cryptol. ePrint Arch. (2022)
7. shing Chen, M., Ding, J., Kannwischer, M., Patarin, J., Petzoldt, A., Schmidt, D., Yang, B.Y.: Rainbow signature: One of the three nist post-quantum signature finalists, https://www.pqcrainbow.org
8. Courtois, N.T., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding. Lecture Notes in Computer Science, vol. 1807, pp. 392–407. Springer (2000). https://doi.org/10.1007/3-540-45539-6_27, https://doi.org/10.1007/3-540-45539-6_27
9. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Applied Cryptography and Network Security (ACNS 2005). Lecture Notes in Computer Science, vol. 3531, pp. 164–175. Springer (2005). https://doi.org/10.1007/11496137_12, https://link.springer.com/chapter/10.1007/11496137_12
10. Faugere, J.C.: A new efficient algorithm for computing gröbner bases (f4). Journal of Pure and Applied AlgebraJournal of Pure and Applied Algebra pp. 61–88 (1999)
11. Faugère, J., Levy-dit-Vehel, F., Perret, L.: Cryptanalysis of minrank. In: Wagner, D.A. (ed.) Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5157, pp. 280–296. Springer (2008). https://doi.org/10.1007/978-3-540-85174-5_16, https://doi.org/10.1007/978-3-540-85174-5_16
12. Furue, H., Ikematsu, Y.: A new security analysis against MAYO and QR-UOV using rectangular minrank attack. In: Shikata, J., Kuzuno, H. (eds.) Advances in Information and Computer Security - 18th International Workshop on Security, IWSEC 2023, Yokohama, Japan, August 29-31, 2023, Proceedings. Lecture Notes in Computer Science, vol. 14128, pp. 101–116. Springer (2023). https://doi.org/10.1007/978-3-031-41326-1_6, https://doi.org/10.1007/978-3-031-41326-1_6
13. Furue, H., Ikematsu, Y., Hoshino, F., Takagi, T., Kosuge, H., Yamakoshi, K., Akiyama, R., Nakamura, S., Orihara, S., Kinjo, K.: Qr-uov: Specification document, round 2. Tech. Rep. Version for Round 2, National Institute of Standards and Technology (NIST) / CSRC (February 2025), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/qruov-spec-round2-web.pdf, accessed: 2025-10-17
14. Ganguly, A., Karmakar, A., Saxena, N.: VDOO: A short, fast, post-quantum multivariate digital signature scheme. In: Chattopadhyay, A., Bhasin, S., Picek, S., Rebeiro, C. (eds.) Progress in Cryptology - INDOCRYPT 2023 - 24th International Conference on Cryptology in India, Goa, India, December 10-13, 2023, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14460, pp. 197–222. Springer (2023). https://doi.org/10.1007/978-3-031-56235-8_10, https://doi.org/10.1007/978-3-031-56235-8_10

15. Goubin, L., Courtois, N.: Cryptanalysis of the ttm cryptosystem. vol. 1976, pp. 44–57 (12 2000). https://doi.org/10.1007/3-540-44448-3_4
16. Guo, H., Jin, Y., Pan, Y., He, X., Gong, B., Ding, J.: Practical and theoretical cryptanalysis of VOX. In: Saarinen, M.O., Smith-Tone, D. (eds.) Post-Quantum Cryptography - 15th International Workshop, PQCrypto 2024, Oxford, UK, June 12-14, 2024, Proceedings, Part II. Lecture Notes in Computer Science, vol. 14772, pp. 186–208. Springer (2024). https://doi.org/10.1007/978-3-031-62746-0_9, https://doi.org/10.1007/978-3-031-62746-0_9
17. Ikematsu, Y., Nakamura, S., Takagi, T.: Recent progress in the security evaluation of multivariate public-key cryptography. IET Information Security **17**(2), 210–226 (2023). https://doi.org/10.1049/ise2.12092, https://doi.org/10.1049/ise2.12092
18. Kipnis, A., Shamir, A.: Cryptanalysis of the hfe public key cryptosystem by relinearization. Advances in Cryptology - CRYPTO 1999, Springer **1666**, 788 (1999)
19. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. Lecture Notes in Computer Science, vol. 1592, pp. 206–222. Springer (1999). https://doi.org/10.1007/3-540-48910-X_15, https://doi.org/10.1007/3-540-48910-X_15
20. Kipnis, A., Shamir, A.: Cryptanalysis of the oil & vinegar signature scheme. In: Krawczyk, H. (ed.) Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 257–266. Springer (1998). https://doi.org/10.1007/BFb0055733, https://doi.org/10.1007/BFb0055733
21. National Institute of Standards and Technology (NIST): Unbalanced Oil and Vinegar (UOV) Specification — Round 1 Submission. Technical Report Spec-Files/Round-1, NIST Computer Security Resource Center, Gaithersburg, MD (May 2023), accessed: 2025-06-07
22. NIST: Round 3 submissions, https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions
23. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) Advances in Cryptology - EUROCRYPT '96, International Conference on the Theory and Application of Cryptographic Techniques, Saragossa, Spain, May 12-16, 1996, Proceeding. Lecture Notes in Computer Science, vol. 1070, pp. 33–48. Springer (1996). https://doi.org/10.1007/3-540-68339-9_4, https://doi.org/10.1007/3-540-68339-9_4
24. Patarin, J.: The oil and vinegar algorithm for signatures. Presented at the Dagstuhl Workshop on Cryptography (September 1997), unpublished manuscript
25. Patarin, J., Cogliati, B., Faugère, J.C., Fouque, P.A., Goubin, L., Larrieu, R., Macario-Rat, G., Minaud, B.: Vox specification v1.0. Tech. rep., NIST Post-Quantum Cryptography Project (June 2023), https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/vox-spec-web.pdf, round 1 Additional Signatures, Digital Signature Schemes
26. Team, M.: Mayo: A round 2 candidate in the nist pqc standardization process. Tech. rep., PQMayo.org (2023), https://pqmayo.org/assets/specs/mayo-round2-white-back.pdf, accessed: 2025-10-17

Params	Rectangular		Rectangular+Simple			
(q, v, d, o_1, o_2)	Avg (sec)	SD (sec)	# Trials	Exp # Trials	Avg (sec)	SD (sec)
(16,4,2,2,2)	0.045	0.008	12.50	16	0.068	0.053
(16,5,2,2,3)	0.629	0.019	10.70	16	0.187	0.192
(16,5,2,3,3)	1.004	0.005	20.50	16	0.357	0.243
(16,2,1,2,2)	0.008	0.004	11.44	16	0.021	0.020
(16,3,2,2,2)	0.020	0.000	7.00	16	0.038	0.040

Table 11. This table reports aggregate statistics regarding the Rectangular MinRank attack and the Combined Combined Simple + RecMin attack from experiments run on the shared GitHub code. Avg (sec) and SD (sec) report the average and standard deviation of that attack's runtime per trial, in CPU seconds. The "# Trials" column reports the average number of random $\mathbf{x}$ vectors needed to be sampled in order for the combined attack to be successful. The "Exp # Trials" reports the predicted average number of $\mathbf{x}$ samples needed to complete the Simple step, which can be computed as $1/p$, where p is defined in Equation (6) (but is often estimated as q). We note that this code is not optimized, is using the Kipnis-Shamir MinRank algorithm, and the code was run on the magma online calculator.

27. Wang, L.C., Tseng, P.E., Kuan, Y.L., Chou, C.Y.: A simple noncommutative UOV scheme. Cryptology ePrint Archive, Paper 2022/1742 (2022), `https://eprint.iacr.org/2022/1742`

A Concrete Security

Table 11 reports experimental results for the attack on toy parameter sets. We ran our Magma implementation using the Magma Web calculator, fixing the random seed to 1 for reproducibility. Using the Kipnis–Shamir modeling implementation, we record aggregate runtime and trial-count statistics for smaller VDOO instances and compare them with the expected behavior. We selected toy parameters satisfying the approximate proportion $d \approx o_1 \approx o_2 \approx m/3$.