VDOO: A Short, Fast, Post-Quantum Multivariate Digital Signature Scheme

Anindya Ganguly, Angshuman Karmakar, and Nitin Saxena

Department of CSE, IIT Kanpur
{anindyag,angshuman,nitin}@cse.iitk.ac.in

Abstract. Hard lattice problems are predominant in constructing post-quantum cryptosystems. However, we need to continue developing post-quantum cryptosystems based on other quantum hard problems to prevent a complete collapse of post-quantum cryptography due to a sudden breakthrough in solving hard lattice problems. Solving large multivariate quadratic systems is one such quantum hard problem.
Unbalanced Oil-Vinegar is a signature scheme based on the hardness of solving multivariate equations. In this work, we present a post-quantum digital signature algorithm VDOO (Vinegar-Diagonal-Oil-Oil) based on solving multivariate equations. We introduce a new layer called the diagonal layer over the oil-vinegar-based signature scheme Rainbow. This layer helps to improve the security of our scheme without increasing the parameters considerably. Due to this modification, the complexity of the main computational bottleneck of multivariate quadratic systems i.e. the Gaussian elimination reduces significantly. Thus making our scheme one of the fastest multivariate quadratic signature schemes. Further, we show that our carefully chosen parameters can resist all existing state-of-the-art attacks. The signature sizes of our scheme for the National Institute of Standards and Technology's security level of I, III, and V are 96, 226, and 316 bytes, respectively. This is the smallest signature size among all known post-quantum signature schemes of similar security.

Keywords: Post-quantum · Digital signature · Multivariate Cryptography · Oil-Vinegar · Multivariate root-finding

1 Introduction

Cryptography is the study of different methods to safeguard our sensitive information in the ever-expanding digital world. The security assurances of cryptographic schemes especially public-key cryptographic schemes emanate from the computational intractability of some underlying hard problems. Currently, public-key cryptographic schemes such as Rivest-Shamir-Adleman [51], elliptic-curve discrete logarithm [44] are predominant in our public-key infrastructure. However, in the context of the rapid development of quantum computers, these schemes exhibit a significant drawback. The underlying hard problems of these schemes *i.e.* integer factorization and discrete logarithm problem can be solved *easily* due to the polynomial time quantum algorithms developed by Shor [54] and Proos-Zalka [50] respectively. Therefore, quantum-resistant hard problems have gained popularity among designers for designing public-key cryptosystems for the future. A landmark event in the development of such quantum-resistant or post-quantum cryptography (PQC) is the PQC standardization procedure [19] initiated by the National Institute of Standards and Technology (NIST) to select quantum-safe cryptographic primitives such as key encapsulation mechanisms (KEM), public-key encryption (PKE), and digital signature algorithm.

In 2022, NIST standardized [3] one KEM (Crystals-Kyber [15]) and three signature schemes (SPHINCS+ [4], Crystals-Dilithium [26], and Falcon [32]) after rigorous scrutiny spanning multiple years. Among these only SPHINCS+ is based on the hardness of cryptographically secure hash functions, while Crystals-Kyber (KEM), Crystals-Dilithium, and Falcon are based on hard lattice problems. As the majority of these constructions are lattice-based, there is a lingering risk that a breakthrough in the cryptanalysis of lattice-based cryptography can reduce the security of these schemes drastically. Thus putting the whole plan to migrate to post-quantum cryptography in jeopardy. Such incidents are not uncommon. Recently, Decru et al. [18] proposed an attack to completely break the security of supersingular isogeny Diffie-Hellman [31] which was earlier considered quantum-safe and was also a finalist in the NIST's standardization procedure. Therefore, it is prudent to diversify the portfolio of different quantum-safe problems for seamless migration to a post-quantum world. There exist other problems that are considered quantum-safe, such as multivariate quadratic (MQ) [46,39], isogeny-based [22], and code-based [8]. Standardizing cryptographic primitives necessitates a rigorous and comprehensive investigation. NIST reissued a call [20] for quantum-safe signature schemes to standardize some more signature schemes to diversify the portfolio of quantum-resistant schemes. Due to its small signature size, multivariate oil-vinegar construction has gained significant attention during this standardization process.

Multivariate cryptography relies on the intractability of root findings of MQ equations. The goal of the MQ problem is to find a solution to a system of multivariate quadratic polynomials in the finite field $\mathbb{F}_q$. In other words, the hardness classification of this problem is NP-hard [38]. Numerous schemes, such as Matsumoto-Imai encryption scheme [43], Oil-Vinegar [46] signature, Rainbow [24] signature, Triangular [45,53,60] signature, Simple Matrix encryption [56], and Mayo [12], have been developed based on multivariate cryptography. Patarin first proposed the Oil-Vinegar signature [46]. A successful forgery attack was shown by Kipnis and Shamir [40] against this scheme. Further, Kipnis, Patarin, and Goubin upgraded the signature scheme by proposing Unbalanced Oil-Vinegar (UOV) [39].

Rainbow was a third-round NIST candidate [24], which is the first multi-layer construction based on unbalanced oil-vinegar. Therefore, the cryptanalysis of Rainbow has been a well-studied area for the last decade. This resulted in many new novel attacks such as direct attack [6,27,28], min-rank attack [14,6,7,5], band-separation attack [25,57,55], rectangular min-rank and intersection attack [10]. In 2023, Beullens proposed a cryptanalysis and reduced the security of Rainbow significantly. Rainbow team suggested using the old SL-3 (high security) parameter set as new SL-1 (low security) parameters [36] to mitigate the attack. As Beullens' attack only applies to the Rainbow structure, therefore building scheme on the top of the oil-vinegar layer is still believed to be secure.

In 2022, Cartor *et al.* internally perturbed the second layer of Rainbow by mixing oil variables quadratically [17]. However, this mixing significantly increased the signature generation time. Also, parameter sets proposed by designers are not practical in terms of efficiency. Therefore, designing a new signature scheme that can resist the simple attack while being practical, is an interesting open problem.

1.1 Our Contribution and Motivation

In the context of this endeavor, we summarize our contributions below.

- We review related multivariate signature schemes and provide a comprehensive analysis of their design and performance in Section 2.
- We present Vinegar-Diagonal-Oil-Oil (VDOO), a novel multivariate signature scheme based on unbalanced oil and vinegar in Section 3. Compared to other UOV schemes VDOO boasts three primary benefits: *simplicity*, *efficiency*, and *security* (see Sections 4 and 5). To the best of our knowledge, we are the first to introduce a diagonal layer within the UOV framework, demonstrating that it enhances efficiency without compromising security.
- We establish that VDOO effectively withstands all current attacks and outline the EUF-CMA security of our scheme. Through meticulous parameter selection, our findings reveal that it achieves a remarkably compact smallest signature size of 96 bytes (see Sections 4 and 5), contrasting favorably with NIST-standardized post-quantum signatures (Crystals-dilithium [26], Falcon [32], and SPHINCS+ [4]) .

Introduction of a new simple design element. VDOO is a new layer-based construction, which has one diagonal layer and then two UOV layers. We are adding each new variable in the central polynomial one by one diagonally. This offers efficiency. This translates to a reduction of the Gaussian elimination ($\mathsf{GE}_{(q,n)}$ [1]) which is the major computational bottleneck in the signature generation process.

Suppose $x_1, x_2, \cdots, x_v, x_{v+1}, \cdots, x_{v+d}, \cdots, x_{v+d+o_1}, \cdots,\ x_{v+d+o_1+o_2=:n}$ are n variables defined over $\mathbb{F}_q$. In our construction, we call first v-variables as *vinegar variables*, next d-variables as *diagonal variables*, then next o_1 variables are *first-layer oil variables*, and last o_2 variables are *second-layer oil variables*. Figure 1 illustrates the distribution of the variables in each layer of the VDOO central polynomial map.

Efficiency. To thwart Beullens' simple attack [11], the authors of Rainbow increased the parameter set [36], which results in increasing the Gaussian elimination cost. The complexity of Gaussian elimination becomes approximately $o_1^3 + o_2^3$ where o_1 and o_2 are number of oil variables of Rainbow [24]. In our scheme, we adapt $d \approx (o_1 + o_2)/3$, $o_1' \approx (o_1 + o_2)/3$, and $o_2' \approx (o_1 + o_2)/3$ as the new parameters. This adjustment results in a Gaussian elimination complexity of around ${o_1}'^3 + {o_2}'^3$. To illustrate, consider the signature generation process for security level one (SL-1) parameters [19]: UOV requires $\mathsf{GE}_{(256,64)}$, Rainbow requires $\mathsf{GE}_{(256,32)}$ and $\mathsf{GE}_{(256,48)}$, while VDOO needs only $\mathsf{GE}_{(16,34)}$ and $\mathsf{GE}_{(16,36)}$ (for further details, refer to Table 2). Consequently, this modification notably improves our scheme's performance.

[1] $\mathsf{GE}_{(q,n)}$: Gaussian elimination on a linear system with n unknowns and n linear equation over $\mathbb{F}_q$.

$$
\begin{array}{rl}
d\text{-diagonal variables} & \left\{\begin{array}{ll}
\boxed{\text{Vinegar } x_1,\cdots,x_v} & \boxed{\text{Diagonal } x_{v+1}} \\
\vdots\ \vdots\ \vdots & \\
\boxed{\text{Vinegar } x_1,\cdots,x_v} & \boxed{\text{Diagonal } x_{v+d}}
\end{array}\right. \\
o_1\text{-oil variables} & \left\{\begin{array}{ll}
\boxed{\text{Vinegar } x_1,\cdots,x_{v+d}} & \boxed{\text{Oil } x_{v+d+1},\cdots,x_{v+d+o_1}} \\
\vdots\ \vdots\ \vdots & \\
\boxed{\text{Vinegar } x_1,\cdots,x_{v+d}} & \boxed{\text{Oil } x_{v+d+1},\cdots,x_{v+d+o_1}}
\end{array}\right. \\
o_2\text{-oil variables} & \left\{\begin{array}{ll}
\boxed{\text{Vinegar } x_1,\cdots,x_{v+d+o_1}} & \boxed{\text{Oil } x_{v+d+o_1+1},\cdots,x_n} \\
\vdots\ \vdots\ \vdots & \\
\boxed{\text{Vinegar } x_1,\cdots,x_{v+d+o_1}} & \boxed{\text{Oil } x_{v+d+o_1+1},\cdots,x_n}
\end{array}\right.
\end{array}
$$

Fig. 1. Variables in each layer of the VDOO central map

Resistance to existing attacks. We comprehensively analyze all possible attacks on multivariate cryptographic schemes against our scheme. In an attempt to recover diagonal variables, potential attackers begin by eliminating the uppermost oil layers. Beullens proposed method [11] facilitates the removal of these layers, aiding attackers. For instance, in order to compromise our round-one parameter set, a straightforward attack necessitates 2^{134}-field operations. Furthermore, Beullens combined this simple attack with the rectangular min-rank attack [10,11]. In line with previous efforts, we execute this combined attack against our scheme, determining that it requires 2^{138}-field operations to break SL-1 parameter set. Additionally, we conduct the intersection attack and the direct attack on our scheme, both of which exhibit complexities exceeding 2^{134}-field operations. Consequently, these references collectively imply that VDOO appears to withstand all known attacks securely. We also outline the EUF-CMA security of the VDOO scheme.

Small signature size. We present multiple parameters that can withstand the aforementioned attacks. Specifically, our level-one parameters that can provide 128-bit classical and 96-bit post-quantum security has a signature size of 96 bytes and public-key size of 238KB (further elaborated in Table 1). This is the smallest signature size among the majority of all multivariate signature schemes (for additional insights, refer to Tables 2 and 3).

Roadmap. In the upcoming Section 2 we present a generic construction of multivariate signatures, and some earlier results. Section 3 proposes a new post-quantum multivariate signature scheme called VDOO. The cryptanalysis of our scheme is presented in Section 4. In Section 5, we give the parameters for different security levels and we also compare our results with the state-of-the-art. Our section 6 presents conclusions and explores potential future directions for our work.

2 Prior Results

In this section, we introduce some essential mathematical notations and symbols. We then provide a generic construction for multivariate signatures. Following that, we outline the central polynomial for UOV and Rainbow [39,13,24]. Additionally, we describe the subspace representation of Rainbow [10], which is particularly valuable for cryptanalysis purposes. Next, we cover recent multivariate signature schemes [12,34,29,23,33] that were submitted as part of the NIST additional round for post-quantum signature standardization [20]. Finally, we present the required hardness assumptions for these multivariate signatures to understand their cryptanalysis.

Notations: Let, $\mathbb{F}_q$ be the finite field with q elements. We define three polynomial maps $\mathcal{S}:\mathbb{F}_q^m \to \mathbb{F}_q^m$, $\mathcal{T}:\mathbb{F}_q^n \to \mathbb{F}_q^n$ and $\mathcal{F}=(f_1,\cdots,f_m):\mathbb{F}_q^n \to \mathbb{F}_q^m$. We denote $[n]$ for the set $\{1,2,\cdots,n\}$ and $[i:j]$ denotes $\{i,i+1,\cdots,j\}$. We use lowercase and bold lowercase alphabets to denote field elements and vectors respectively.

2.1 Generic Multivariate Signature Schemes

Here we briefly describe a generic construction for multivariate signature schemes. Due to the NP-hardness of inverting a randomly generated quadratic system [38]. However, signers can leverage a specially structured quadratic system to efficiently perform the inversion. This specialized system is commonly referred to as the *central map* and is typically denoted as $\mathcal{F}=(f_1,\cdots,f_m)$, where each f_i represents a specifically structured multivariate quadratic polynomial. Signers must conceal this unique structure from third parties to prevent forgery attacks. To achieve this objective, signers employ one or two random invertible linear maps: $\mathcal{S}$ and $\mathcal{T}$ Consequently, the public key is constructed by composing these linear maps along with the central map, denoted as $\mathcal{P}= \mathcal{S}\circ\mathcal{F}\circ\mathcal{T} : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^m$.

The secret key comprises $\mathcal{S}$, $\mathcal{T}$ and $\mathcal{F}$. A hash function, denoted as $\mathcal{H}:\{0,1\}^* \longrightarrow \mathbb{F}_q^m$, is employed to generate a vector $\mathbf{m}\in\mathbb{F}_q^m$ from a message $msg\in\{0,1\}^*$. The signature generation process unfolds as follows: first, compute $\mathbf{d}\leftarrow\mathcal{S}^{-1}(\mathbf{m})$, then $\mathbf{d}'\leftarrow\mathcal{F}^{-1}(\mathbf{d})$, and finally $\mathbf{s}\leftarrow\mathcal{T}^{-1}(\mathbf{d}')$. The signer sends the signature $\mathbf{s}$ for the message msg to the verifier. The verifier simply evaluates the polynomial map $\mathcal{P}$ on $\mathbf{s}$ and checks whether it matches the hash of the message, i.e., whether $\mathbf{m}=\mathcal{P}(\mathbf{s})$ holds or not.

2.2 Unbalanced Oil-Vinegar (UOV)

The Oil-Vinegar (OV) signature scheme was initially introduced by Patarin [46]. However, due to the Kipnis-Shamir's [40] *invariant subspace* attack, this scheme was modified by increasing the number of vinegar variables. This is known as the Unbalanced Oil-Vinegar (UOV) signature scheme [39].

Consider the OV central map, denoted as $\mathcal{F}$. Split all variables of $\mathbf{x}=(x_1,\cdots,x_v,\cdots,x_n)$ into two buckets: the first bucket has first v variables representing vinegar, and the second bucket contains next o variables representing oil, where $n=v+o$ and $o=m$. To create a multivariate quadratic homogeneous polynomial, combine variables involving vinegar $\times$ vinegar and vinegar $\times$ oil, while excluding all oil $\times$ oil terms.

Definition 1 (OV Central Polynomial Map). *A central map $\mathcal{F}=(f_1,\cdots,f_m)$: $\mathbb{F}_q^n \to \mathbb{F}_q^m$ is known as OV central polynomial map when each f_i is of the form $f_i(\mathbf{x}) = \sum_{i=1}^{v}\sum_{j=1}^{n} \alpha_{i,j}^{(k)}\, x_i x_j$ where $i\leq j$, $k\in[v+1:n]$, $\mathbf{x}\in\mathbb{F}_q^n$, and $\alpha_{i,j}^{(k)}\in_U\mathbb{F}_q$.*

Notably, if anyone randomly fixes vinegar variables, then the remaining part would be linear in the oil variables. Therefore, the quadratic system reduces to a linear system of o linear equations with o unknowns.

2.3 Rainbow

Rainbow is a multi-layer variant of UOV [24]. For simplicity consider a two-layer Rainbow. Suppose $n = v + o_1 + o_2$, where the first v variables are vinegar and the next o_1 and o_2 variables are the first and second layer of oil variables respectively. This can be viewed as a UOV map with $v + o_1$ variables and o_1 oil variables and the next layer $v + o_1 + o_2$ variables and o_2 oil variables.

Definition 2 (Rainbow Central Polynomial Map). *The mathematical expression for l-layer Rainbow central polynomial is as follows.*

$$f_k(x_1, x_2, \cdots, x_n) = \sum_{i,j \in [r];\ i \leq j} \alpha_{ij}^{(k)} x_i x_j + \sum_{i \in [r];\ j \in [r+1:r+o_r]} \beta_{ij}^{(k)} x_i x_j$$

where for each $k \in [r+1 : r+o_r]$, elements $\alpha_{ij}^{(k)}$, and $\beta_{ij}^{(k)}$ are taken from $\mathbb{F}_q$; and r denotes the layer.

2.4 Beullens Subspace Description

For a better view of cryptanalysis on Rainbow, Beullens explained the construction of Rainbow via subspaces [10]. Using this description, he derived the simple attack [11]. To elaborate this idea, initially, we define a differential polar form of a polynomial map.

The *differential polar map* of a polynomial map $\mathcal{P}$ is denoted by $\mathcal{DP} : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^m$ and defined as $\mathcal{DP}(\mathbf{x}, \mathbf{w}) = \mathcal{P}(\mathbf{x}+\mathbf{w}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{w})$.

$$\begin{tikzcd}
\mathbb{F}_q^n \arrow[d, "\mathcal{P}"'] & \supset & O_1 \arrow[d, "\mathcal{P}"'] \arrow[dll, "\mathcal{DP}(\mathbf{x},.)"'] & \supset & O_2 \arrow[d, "\mathcal{P}"'] \arrow[dll, "\mathcal{DP}(\mathbf{x},.)"'] & \supset \cdots \supset & O_{l-1} \arrow[d, "\mathcal{P}"'] & \supset & O_l \arrow[d, "\mathcal{P}"] \arrow[dll, "\mathcal{DP}(\mathbf{x},.)"'] \\
\mathbb{F}_q^m & \supset & Q_1 & \supset & Q_2 & \supset \cdots \supset & Q_{l-1} & \supset & Q_l = \{0\}
\end{tikzcd}$$

Fig. 2. l layer Rainbow

Trapdoor information. This part describes the trapdoor information of l-layer Rainbow. At first, signer chooses a secret chain of nested subspaces: input subspaces $O_1 \supset O_2 \supset \cdots \supset O_l$ and output subspaces $Q_1 \supset Q_2 \supset \cdots \supset Q_l = \{0\}$. Using this secret, one can construct a public polynomial map as follows.

- $\mathcal{P}$ maps each O_i to Q_i and
- for any $\mathbf{x} \in_U \mathbb{F}_q^n$, $\mathcal{DP}_\mathbf{x} : O_i \to Q_{i-1}$ is a linear map (see Figure 2)

Inversion. In this methodology, the goal is to compute $\mathbf{x} \in \mathbb{F}_q^n$ from given $\mathbf{y} \in \mathbb{F}_q^m$ such that $\mathbf{y} = \mathcal{P}(\mathbf{x})$. The knowledge of nested sequences of input and output subspaces is used in this computation. At first glance, for l-layer Rainbow, the value of the unknown $\mathbf{x}$ can be represented as $\mathbf{v} + \mathbf{o_1} + \cdots + \mathbf{o}_l$ where all of the $\mathbf{o_i} \in O_i$. Fix $\mathbf{v} \in_U \mathbb{F}_q^n$. Then $\mathcal{P}$ is used in conjunction with the ith-layer's output subspace Q_i to calculate $\mathbf{o}_i$. For the sake of clarity, let's define the *quotient space* $\overline{O}_i := O_i / O_{i+1}$.

Using the knowledge of sequences of subspaces, the goal is to find $\mathbf{o}_i$ for all i. This will lead to computing the preimage of any element from $\mathbb{F}_q^n$. For computing $\overline{\mathbf{o}}_i \in \overline{O}_i$, use the following relation (note that, from definition, $\mathcal{P}\ (\overline{\mathbf{o}}_i) = 0$),

$$\begin{aligned} &\mathcal{P}(\mathbf{v} + \overline{\mathbf{o}}_i) + Q_i = \mathbf{y} + Q_i \\ \Longrightarrow\ &\mathcal{P}(\mathbf{v}) + \mathcal{P}\ (\overline{\mathbf{o}}_i) + \mathcal{DP}\ (\mathbf{v}, \overline{\mathbf{o}}_i) + Q_i = \mathbf{y} + Q_i. \end{aligned}$$

Earlier $\mathbf{v}$ is fixed, so the quadratic system reduces to a linear system. The number of constraints and variables are the same for the linear system. This implies that a unique solution can be obtained with probability $(1 - \frac{1}{q})$. Repeatedly running this procedure, one can compute all $\mathbf{o}_i$, which implies that preimage $\mathbf{x}$ will be computed.

In 2022, Beullens [11] reduced the security level of Rainbow. He showed for small $n - m$, recovering all subspaces is significantly efficient. Also, the small finite field size accelerates the attack.

2.5 Concurrent Proposals

The NIST additional signature submission call [20] received a total of eleven multivariate signature schemes e.g. Mayo [12], QR-UOV [34], TUOV [23], etc. Most of them are based on the old *unbalanced Oil-Vinegar* structure. For example, Mayo [12] employed a UOV structure along with a new *whipped-up MQ* (WMQ) approach. QR-UOV is another variant of UOV where the public key is represented by block matrices, with each element corresponding to an element in a quotient ring [34]. Also, in 2022, a new proposal, called IPRainbow [17] was made by perturbing the central polynomials of the second layer by s variables. This change although decreases the attack probability by $1/q^s$, the running time significantly increases due to the usage of Gröbner basis technique for inversion.

2.6 Hardness of Multivariate Cryptography

Here, we describe other approaches used in the cryptanalysis of multivariate signatures apart from the direct solution of MQ equations.

1. **Min-rank.** Let $M_1, M_2, \cdots, M_k \in \mathbb{F}_q^{n \times m}$ be the given matrices and $r \in \mathbb{N}$, find a non-trivial linear combination (with $m_1, m_2, \cdots, m_k \in \mathbb{F}_q$) so that rank $\left(\sum_{i=1}^k m_i M_i\right) \leq r$. This problem is called the *min-rank* problem and has been shown to be NP-hard [16]. The min-rank problem appeared as a cryptanalytic tool in multivariate cryptography [41,30,6,10]. This attack helps to find a linear combination of public matrices which sums up to a low-rank matrix.

2. **EIP.** Find an equivalent composition of $\mathcal{P}=\mathcal{S}'\circ\mathcal{F}'\circ\mathcal{T}'$, where $\mathcal{S}'$ and $\mathcal{T}'$ are equivalent affine maps, and $\mathcal{F}'$ is an equivalent central map. The above problem is the *Extended Isomorphism of Polynomials (EIP)* problem. No such hardness classification is known (though it subsumes graph isomorphism problem [1,2]), but for some instances, polynomial time algorithms exist [40].

3 Our Proposal: VDOO Signature Scheme

In our scheme, we introduce a new design element called *diagonals* into the Oil-Vinegar scheme. Let, $\mathbf{x}\in\mathbb{F}_q^n$, we pick the first v variables as vinegar variables. We denote the next d variables as diagonal variables. In this layer, we introduce d quadratic equations. In any i-th $(1\leq i\leq d)$ equation, only $v+i$-th variable is unknown among $v+i$ variables. In the following layers, we apply the Oil-Vinegar technique. This means we can generate o_1 OV polynomials using $v+d$-vinegar variables and newly added o_1-oil variables. Further, we construct o_2 OV polynomials using $v+d+o_1$-vinegar variables and newly added o_2-oil variables. Finally, we have a quadratic system with $n=v+d+o_1+o_2$ variables and $m=d+o_1+o_2$ homogeneous quadratic equations.

3.1 VDOOSetUp: Generate Parameters

To construct polynomial maps we need to define parameters associated with this. In this phase algorithm takes input the security parameter λ and output the parameter tuple, that is $\mathsf{params}=(q,\, v,\, d,\, o_1,\, o_2\,)\leftarrow\mathsf{VDOOSetUp}(1^\lambda)$. Here,

- Finite field $\mathbb{F}_q$ which has q elements.
- Positive integers v, d, o_1, and o_2, where v denotes the number of vinegar variables, d is the number of diagonal variables, o_1 and o_2 stands for the number of first and second layer oil variables respectively. Therefore, total number of variables is $n=v+d+o_1+o_2$, and number of equations is $m=d+o_1+o_2$.

To generate the parameter set, signer can use VDOOSetUp algorithm 1

Algorithm 1 VDOOSetUp

Require: Security parameter λ
Ensure: Parameter tuple $\mathsf{params}=(q,v,d,o_1,o_2)$
1: Generate a parameter tuple corresponding to the security level λ
2: **Return** params

3.2 VDOO Central Polynomial Map and Inversion.

Construction of central polynomial map $\mathcal{F}:\mathbb{F}_q^n\rightarrow\mathbb{F}_q^m$ plays an important role in the multivariate signature schemes. To the best of our knowledge, we are the first to propose a central polynomial map that involves vinegar, diagonal, and oil variables in a three-layer construction.

- **Diagonal Layer.** Here, we explain the structure of any central polynomial f_k for the diagonal layer $k \in [v+1 : v+d]$. Each f_k is defined as follows.

$$f_k(x_1,x_2,\cdots,x_n) = \sum_{i=1}^{k-1} \alpha_{i,k}^{(k)} x_i x_k + \sum_{i,j=1, i \leq j}^{k-1} \beta_{i,j}^{(k)} x_i x_j$$

Each coefficient $\alpha_{ij}^{(k)}$, and $\beta_{ij}^{(k)} \in_U \mathbb{F}_q$. The subroutine $\mathsf{DiagPoly}(q,k)$ is used to generate such central polynomial f_k in the diagonal layer.
- **First Oil Layer.** In this oil layer, we use $v+d$ variables as vinegar variables and next o_1 variables as oil variables. All these variables help us to construct o_1 homogeneous quadratic polynomials of the following form.

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

where $k \in [v+d+1 : v+d+o_1]$, $\alpha_{ij}^{(k)}$, and $\beta_{ij}^{(k)} \in_U \mathbb{F}_q$.
- **Second Oil Layer.** The topmost oil layer has $v+d+o_1$ vinegar and o_2 oil variables. That means, it has o_2 quadratic equations. Those equations are of the form

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

where $k \in [v+d+o_1+1 : v+d+o_1+o_2 = n]$ and $\alpha_{ij}^{(k)}$, and $\beta_{ij}^{(k)} \in_U \mathbb{F}_q$. We denote this as $\mathsf{OVPoly}(q,\ v,\ o)$ to generate o central polynomials.

Here, Algorithm 2, uses OVPoly and DiagPoly to generate a VDOO central map $\mathcal{F}$.

Algorithm 2 VDOOCentPoly

Require: Parameter tuple $params = (q,v,d,o_1,o_2)$
Ensure: Central map $\mathcal{F} = (f_1,\cdots,f_m) : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$
1: Compute $m = d+o_1+o_2$ and $n = v+m$.
2: **for** $1 \leq i \leq d$
3: $\quad f_i \leftarrow$ DiagPoly (q,i)
4: **for** $d+1 \leq i \leq d+o_1$
5: $\quad f_i \leftarrow$ OVPoly $(q, v+d, o_1)$
6: **for** $d+o_1+1 \leq i \leq m$
7: $\quad f_i \leftarrow$ OVPoly $(q, v+d+o_1, o_2)$
8: **Return** VDOO central polynomial $\mathcal{F}$

Inversion. The main computational bottleneck of UOV-based constructions is the inversion of the central polynomial. It requires Gaussian elimination which runs in $O(N^3)$. However, in our scenario inversion of the diagonal polynomials is straightforward as there is only one unknown variable. Nevertheless, the inversion of OV polynomials in the remaining two layers each needs a Gaussian elimination. Therefore, inverting VDOO central polynomial map needs two Gaussian elimination only. This is shown in Algorithm 3. Here, the subroutine ST fixes $x_1,\cdots,x_l$ in f_i and convert it to $\tilde{f}_i$ for all i. The $\mathsf{GE}_{(q,l)}$ denotes Gaussian elimination for l unknowns over the linear system of equations $(\ \tilde{f}_i = y_i\ )_{i=1}^{l}$. The function GE returns a failure when the rank of the matrix representing the linear system is less than l.

Algorithm 3 $\mathsf{VDOOCentPoly\_Inversion}$

Require: Central map: $\mathcal{F} = (f_1,\cdots,f_m) : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$ and $\mathbf{y} \in \mathbb{F}_q^m$, and params.
Ensure: A vector $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{y}$.

$m \leftarrow d + o_1 + o_2$ and $n \leftarrow v + m$
Randomly fix first v-vinegar variables $x_1,\cdots,x_v \leftarrow_{\$} \mathbb{F}_q$
for $1 \le i \le d$
  compute x_{v+i} using $y_i, x_1, \cdots, x_{v+i-1}$ and f_i.
$(\ \tilde{f}_{d+1},\cdots,\tilde{f}_{v+d}) \leftarrow \mathsf{ST}\big(f_{d+1}(x_1,\cdots,x_{v+d})\ ,\cdots,\ f_{d+o_1}(x_1,\cdots,x_{v+d})\big)$
$(x_{v+d+1},\cdots,x_{v+d+o_1}) \leftarrow \mathsf{GE}_{(q,o_1)}(\tilde{f}_{d+1} = y_{d+1},\cdots,\tilde{f}_{d+o_1} = y_{d+o_1})$.
$(\tilde{f}_{d+o_1+1},\cdots,\tilde{f}_m) \leftarrow \mathsf{ST}\big(f_{d+o_1+1}(x_1,\cdots,x_{n-o_2}),\cdots,f_m(x_1,\cdots,x_{n-o_2})\big)$
$(x_{v+d+o_1+1},\cdots,x_n) \leftarrow \mathsf{GE}_{(q,o_2)}(\tilde{f}_{d+o_1+1} = y_{d+o_1+1},\cdots,\tilde{f}_m = y_m)$
Return $\mathbf{x} \in \mathbb{F}_q^n$

3.3 $\mathsf{VDOOKeyGen}$: VDOO Key Generation

The $\mathsf{VDOOKeyGen}$ in Algorithm 4 generates two random invertible affine maps $\mathcal{S} : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^m$ and $\mathcal{T} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ along with the VDOO-central map $\mathcal{F} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$. Here, *secret/signing key* is $\mathcal{S}$, $\mathcal{F}$, and $\mathcal{T}$ and *public/verification key* is the composition map $\mathcal{P}$, where $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m$. Note that, the individual information of secret maps allows user to compute the inverse of $\mathcal{P}$ efficiently. We denote $S \leftarrow \mathsf{randomMatrix}\,(q, m, \mathit{seed})$ to generate a random $m \times m$ matrix over $\mathbb{F}_q$ from a *seed*, $\mathsf{invMat}\,(q, m, S)$ helps to compute the inverse of a $m \times m$ matrix S over $\mathbb{F}_q$, and $\mathsf{Affine}(S,\mathbf{a})$ computes $\mathcal{S} \leftarrow S \cdot \mathbf{x} + \mathbf{a}$.

3.4 $\mathsf{VDOOSign}$: VDOO Signature Generation

Similar to the other OV based constructions [12,23,39,24], we use the hash-and-sign paradigm for our signature algorithm as shown in Algorithm 5. We use a hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^m$. Signer knows each polynomial map, so it can compute the inverse of each map *i.e.* $\mathcal{S}^{-1}$, $\mathcal{F}^{-1}$, $\mathcal{T}^{-1}$. If GE reports a failure during the computation of $\mathcal{F}^{-1}$, we restart the process by regenerating the salt and repeating the entire procedure. Finally, the signature is computed as $\mathcal{P}^{-1}(\mathcal{H}(\mathcal{H}(msg)||salt))$.
Efficiency analysis. As mentioned earlier, the major computational overhead of OV-based schemes is the Gaussian elimination procedure. In VDOO, during signing,

Algorithm 4 VDOOKeyGen

Require: Parameter tuple params.
Ensure: Generate public and private key pair.
– Public key: $\mathsf{pk} = \mathcal{P}$.
– Secret key: $\mathsf{sk} = \mathcal{S}$, $\mathcal{T}$, and $\mathcal{F}$.

$m \leftarrow d + o_1 + o_2$ and $n \leftarrow m + v$
$seed \leftarrow PRNG(1^\lambda)$ ▷ λ is the security parameter
while $(\det(S) \neq 0 \;\&\&\; \det(T) \neq 0)$ **do**
  $S \leftarrow \mathsf{randomMatrix}\ (q, m, seed)$ ▷ $S \in_U \mathbb{F}_q^{m \times m}$
  $T \leftarrow \mathsf{randomMatrix}\ (q, n, seed)$ ▷ $T \in_U \mathbb{F}_q^{n \times n}$
end while
$\mathbf{a} \in_U \mathbb{F}_q^m$ and $\mathbf{b} \in_U \mathbb{F}_q^n$ ▷ generate two random vector
$invS \leftarrow \mathsf{invMat}(q, m, S)$ and $invT \leftarrow \mathsf{invMat}(q, n, T)$ ▷ compute inverse of matrices
$\mathcal{S} \leftarrow \mathsf{Affine}(S, \mathbf{a})$ and $\mathcal{T} \leftarrow \mathsf{Affine}(T, \mathbf{b})$ ▷ Constructing invertible affine maps
$\mathcal{F} \leftarrow \mathsf{VDOOCentPoly}(\mathsf{params})$ ▷ generate VDOO central map
Compute $\mathcal{P} \leftarrow \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$
Return $\mathsf{pk} = \mathcal{P}$ and $\mathsf{sk} = (invS, \mathbf{a}, invT, \mathbf{b})$ (equivalently sending $\mathcal{S}$, and $\mathcal{T}$).

Algorithm 5 VDOOSign

Require: $\mathsf{sk} = (invS, \mathbf{a}, invT, \mathbf{b})$, message msg, and $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$
Ensure: a signature $\sigma = (\mathbf{s}, salt)$

$salt \longleftarrow PRNG$
Use hash function $\mathbf{d} \leftarrow \mathcal{H}(\mathcal{H}(msg) || salt)$
Compute $\mathbf{t} = invS \times (\mathbf{d} - \mathbf{a})$ ▷ $\mathbf{t} = \mathcal{S}^{-1}(\mathbf{d})$
Compute $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{t})$ using VDOOCentPoly_Inversion 3.
Compute $\mathbf{s} = invT \times (\mathbf{y} - \mathbf{b})$ ▷ $\mathbf{s} = \mathcal{T}^{-1}(\mathbf{y})$
Return signature $\sigma = (\mathbf{s}, salt)$

we have to compute only one Gaussian elimination *i.e.* computation of $\mathcal{F}^{-1}$. The computation of $\mathcal{S}^{-1}$ and $\mathcal{T}^{-1}$ can be done during the key-generation procedure. In VDOO the computation of $\mathcal{F}^{-1}$ is also very efficient compared to other OV-based schemes as the number of unknowns is smaller in VDOO as shown in Table 2.

3.5 VDOOVerif: VDOO Verification

Our verification procedure is simple. It needs a polynomial evaluation of $\mathcal{P}$, requiring just $O(N^3)$ field operations. Compute $\mathbf{d}' = \mathcal{P}(\mathbf{s})$ from public key $\mathcal{P}$ and signature $\sigma = (\mathbf{s}, salt)$ The signatures is accepted if $\mathbf{d}' = \mathcal{H}(\mathcal{H}(msg) \;||\; salt)$ holds, else rejected.

3.6 Key Size Computation

Our VDOO contains one diagonal layer and two UOV layers. The size of the private key is determined first, followed by the size of the public key.

Algorithm 6 VDOOVerif

Require: $\mathsf{pk}=\mathcal{P}$; message msg; signature $\sigma=(\mathbf{s},salt)$ and $\mathcal{H}:\{0,1\}^*\to\mathbb{F}_q^m$.
Ensure: *accept* or *reject*
1: Use hash function to compute $\mathbf{d}\leftarrow\mathcal{H}(\mathcal{H}(msg)\ ||\ salt)$
2: Compute $\mathbf{d}'=\mathcal{P}(\mathbf{s})$
3: **if** $\mathbf{d}=\mathbf{d}'$ **then** output *accept*
4: **else** *reject*
5: **end if**
6: **Return** *accept* or *reject*

– Size of the central map $\mathcal{F}$ for a diagonal layer having depth d is $\sum_{i=1}^{d}\left(\frac{v_i(v_i+1)}{2}+v_i\right)$ field elements.
– Size of the central map $\mathcal{F}$ for a UOV layer is around $o\times\left(\frac{v(v+1)}{2}+ov\right)$ field elements. Such UOV layer has v vinegar variables and o oil variables.

The first diagonal layer has $v_1=n-m$ vinegar variables. In any diagonal layer, a central polynomial f_i has v_i vinegar variables and f_{i+1}-th polynomial has $v_{i+1}=v_i+1$ vinegar variables. The sizes of the two affine transformations are as follows: for $\mathcal{S}$ we need $m(m+1)$, while for $\mathcal{T}$ we need $n(n+1)$, field elements. These maps can be generated using a random seed.

Now we are interested in computing the size of the public key of standard VDOO. Each n-variate quadratic polynomial requires $\frac{(n+1)(n+2)}{2}$ field elements. Therefore, the size of the public key is $m\frac{(n+1)(n+2)}{2}$. Further optimization of public key is possible [48,49]. It optimized the public key size from $O(mn^2\log q)$ to $O(m^3\log q)$.

3.7 Subspace Description of VDOO Central Polynomial

Our scheme can be explained through Beullens's subspace descriptions [10]. This description is useful to understand the cryptanalysis of VDOO. In this case, we have $d+2$ input and output subspaces. These sequences of nested subspaces are as follows.

– **Input subspaces** $\mathbb{F}_q^n\supset D_1\supset D_2\supset\cdots\supset D_d\supset O_1\supset O_2$.
– **Output subspaces** $\mathbb{F}_q^m\supset Q_{1,1}\supset Q_{1,2}\supset\cdots\supset Q_{1,d}\supset Q_2\supset Q_3=\{0\}$.

In the Figure 3 (single arrow denotes $\mathcal{P}$ and bold arrow denotes $\mathcal{DP}(\mathbf{x},\cdot)$), these following relations will hold: $\dim(D_i)=\dim(D_{i+1})+1$ and $\dim(Q_{1,i})=\dim(Q_{1,i+1})+1$ for $1\le i<d$. Also, $\dim(D_1)=m$, $\dim(D_i)=\dim(Q_{1,i-1})$ for $1<i\le d$. In addition, $\dim(O_1)=\dim(Q_{1,d})=o_1+o_2$, $\dim(O_2)=\dim(Q_2)=o_2$.

The signer first fixes $\mathbf{v}\in_U\mathbb{F}_q^n$. Since $\dim(\tilde{D}_i)=\dim(D_i)-\dim(D_{i+1})=1$, so for diagonal layer computing $\mathbf{d}_1,\cdots,\mathbf{d}_d$ is very easy. Once these vectors are found, then update $\mathbf{v}\leftarrow\mathbf{v}+\mathbf{d}_1+\cdots+\mathbf{d}_d$. Now, signer needs to solve for $\tilde{\mathbf{o}}_1\in\tilde{O}_1(=O_1/O_2)$, so that the following relation holds. Note that, $\dim(\tilde{O}_1)=o_1$.

$$\mathcal{P}(\mathbf{v})+\mathcal{DP}(\mathbf{v},\tilde{\mathbf{o}}_1)=\mathbf{t}\mod Q_2.$$

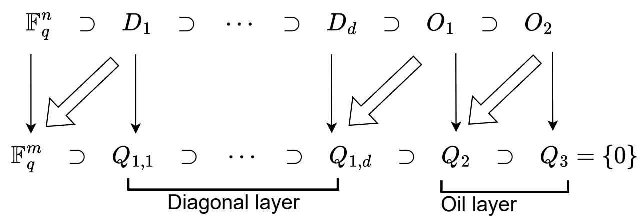

Fig. 3. Central polynomial of VDOO

We know that the above equation is a linear system of o_1 variables and o_1 equations. With the probability $(1-1/q)$, the signer will able to compute $\mathbf{o}_1$. Then signer again updates $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{o}_1$ and follow a similar strategy to find $\mathbf{o}_2 \in O_2$. Thus the signer can finally compute the pre-image of $\mathbf{t}$.

4 Security Analysis of VDOO

Cryptanalysis that targets solving the MQ problem directly, is known as the direct attack in multivariate cryptography [6,27,28,9]. Later researchers have used the special structure of the quadratic system and improved the state-of-the-art, like, band-separation attack [25,57,55], intersection attack [10], and simple attack [11].

To determine the complexity of the attacks described below by the number of field multiplications required to perform the attack. One $\mathbb{F}_q$-field multiplication needs $(2(\log_2 q)^2 + \log_2 q)$ gates. Here, each $2(\log_2 q)^2$-bit stands for one $(\log_2 q)^2$-bit multiplication (represented as AND gates) and the same number of additions (represented as XOR gates) during one $\mathbb{F}_q$-multiplication. Additionally, $\log_2 q$ bits are needed for $\log_2 q$-bit additions involved in one $\mathbb{F}_q$-addition, which is required for each field multiplication that occurs during an attack. For example, the cost one $\mathbb{F}_{16}$-multiplication requires 36 gates. Such a strategy to determine the complexity is standard and has been also followed in other MQ-based signature schemes [34,12,29].

Henceforth, in this document, we use the parameter set $(q,v,d,o_1,o_2) = (16,60,30,34,36)$ as an example to demonstrate the complexity of the following attacks. Incidentally, this is also our SL-1 parameter. Our full parameter set is given in Table. 1.

4.1 Direct Attack on VDOO

The direct attack is the fundamental methodology for forging any multivariate signature scheme. To counterfeit a VDOO signature, an attacker aims to solve an underdetermined system with n variables and m homogeneous equations $(n > m)$, to find $\mathbf{s}$ such that $\mathcal{P}(\mathbf{s}) = \mathbf{t}$. The basic approach involves converting this underdetermined system into a determined one by fixing $n-m$ variables. Subsequently, quadratic system-solving techniques like the Wiedemann XL algorithm [58,21] or Gröbner basis methods such as F4 or F5 [27,28] are applied. Another approach named hybrid approach [9] involves guessing k variables prior to solving the system. The time complexity of this attack, using the approach outlined in [9], is expressed in terms of field multiplications as:

$$\min_{0\leq k\leq m} q^k \cdot 3 \cdot \binom{m-k+d}{d}^2 \binom{m-k}{2}$$

Here, k denotes the number of variables fixed during the algorithm, and d represents the smallest integer for which the coefficient of t^d in the series $\frac{(1-t^2)^m}{(1-t)^{m-k}}$ is non-positive.

Example for SL-1 parameters. Our level one parameter set has 160 variables and 100 constraints. According to [9], we fix 60 variables. Now in the algorithm, if we fix twelve variables, then the value of d is 28. The total complexity is around 2^{280}.

4.2 Simple Attack on VDOO

In 2022, Beullens proposed the *simple attack* against Rainbow [24]. For Rainbow, this highly effective attack reduces n-unknown and m-constraints in the quadratic system to $n-m$-unknown and m-constraints. Now an attacker can apply the same methodology on VDOO to recover the secret key. Recall from Figure. 3, $\mathcal{P}$ is the public polynomial map, and sequences of nested input and output subspaces are,

- ***Input subspaces*** $\mathbb{F}_q^n \supset D_1 \supset D_2 \supset \cdots \supset D_d \supset O_1 \supset O_2$.
- ***Output subspaces*** $\mathbb{F}_q^m \supset Q_{1,1} \supset Q_{1,2} \supset \cdots \supset Q_{1,d} \supset Q_2 \supset Q_3 = \{0\}$.

The main crux of the simple attack lies in finding a vector within O_2 (as depicted in Figure. 3). To achieve this, the attacker must solve a quadratic system with $n-m$ unknowns and m constraints using the XL algorithm. This computational step constitutes the most significant component of the entire attack. Here is a step-by-step outline detailing the cryptanalysis of our scheme using the simple attack.

Input: Public polynomial map $\mathcal{P}$.
Output: Recover sequences of subspaces.
Find a vector $\mathbf{o}\in O_2$: Choose $\mathbf{v}\in_U \mathbb{F}_q^n$. Then from Figure. 3, $\mathcal{DP}_{\mathbf{v}} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is a linear map, in particular it maps O_2 to Q_2. The attacker uses this linear relation to reduce the number of unknowns present in the quadratic system. Therefore, to find a vector, an attacker should solve the following system.

$$\begin{aligned} \mathcal{DP}_{\mathbf{v}}(\mathbf{o}) &= 0 \\ \mathcal{P}(\mathbf{o}) &= 0 \end{aligned}$$

With probability $\approx 1/q$, the attacker successfully guesses a vector in O_2. Later, the attacker deploys the XL algorithm to solve the quadratic system of $n-m$-unknowns and m-constraints. Thus attacker recovers $\mathbf{o}$.

Recover Q_2: Attacker will retrieve Q_2 using the information $\mathbf{o} \in O_2$. Note that, $\mathcal{DP}_{\mathbf{o}} : O_2 \to Q_2$ is a linear map. Therefore,

$$\mathrm{Span}\{\ \mathcal{DP}_{\mathbf{o}}(\mathbf{e}_1), \cdots, \mathcal{DP}_{\mathbf{o}}(\mathbf{e}_n)\ \} \subseteq Q_1$$

for some linearly independent vectors $\mathbf{e}_i$. For enough such $\mathbf{e}_i$'s equality will hold.

Recover O_2: To recover O_2, solve the following system of linear equations. Because with high probability kernel of $\mathcal{DP}_{\mathbf{o}}$ matches with O_2.

$$\begin{aligned}\mathcal{DP}_{\mathbf{o}}(\mathbf{e}_1) &\equiv 0 \mod Q_2\\ \mathcal{DP}_{\mathbf{o}}(\mathbf{e}_2) &\equiv 0 \mod Q_2\\ &\vdots\\ \mathcal{DP}_{\mathbf{o}}(\mathbf{e}_n) &\equiv 0 \mod Q_2\end{aligned}$$

Recover a vector $\mathbf{o}' \in O_1$: Now the quadratic system $\mathcal{P}$ reduces to $m' = m - o_2$ equations and $n' = n - o_2$ variables. To recover O_1, the goal of the attacker is to find a vector in $\mathbf{o} \in O_1$. Again attacker will guess a vector $\mathbf{v}' \in \mathbb{F}_q^{n'}$. Like above, a similar argument shows that $\mathcal{DP}_{\mathbf{v}'}: O_1 \to Q_{1,1}$ is a linear map and the attacker tries to solve the following systems mod Q_2.

$$\begin{aligned}\mathcal{DP}_{\mathbf{v}'}(\mathbf{o}') &= 0 \mod Q_2\\ \mathcal{P}(\mathbf{o}') &= 0 \mod Q_2\end{aligned}$$

The attacker runs the XL algorithm to solve the quadratic system of $n' - m'$-unknowns and m'-constraints.

Recover O_1: Attacks follows same approach as recovering O_2 to recover O_1. Here, an attacker solves a system $\mathcal{DP}_{\mathbf{o}'}(\mathbf{e}'_i) \equiv 0 \mod Q_1$ for $i \leq n'$.

Recovering vectors from diagonal layer: The only task that remains is to find all the diagonal vectors. The attacker can apply Wolf et al.'s [59] trick to find all the diagonal vectors in the layer. Here observe that the computation of finding a vector in O_2, dominates the computation of finding a vector in O_1.

Attack Complexity. The complexity of the first steps dominates the complexity of other steps involved in this algorithm. Basically, a system of n variables and m non-linear equations reduces to a system of m homogeneous equations with $n-m$ variables. This computation can be performed via XL algorithm and it requires

$$3 \cdot q \binom{n-m-1+d}{d}^2 \binom{n-m-1}{2}$$

field operations, where d is the operating degree of the algorithm. It means, d-is the smallest positive integer so that the coefficient of t^d in the power series $(1-t^2)^m/(1-t)^{n-m}$ is non-positive.

Example for SL-1 parameters. Apply Beullens' trick to guess a vector in O_2, which happens with probability $1/q$. Finding one vector on O_2 asks to solve a quadratic system of 100-variables 60-unknowns. This computation is the most costly in the entire algorithm. Solving this quadratic system needs 2^{134} field operations. The guessing needs $1/q$ search and cost of one $\mathbb{F}_{16}$- multiplication needs 36 gates. Therefore, this parameter set provides approximately at-least 128-bit security.

4.3 Rectangular Min-rank Attack on VDOO

Rectangular min-rank attack is proposed by Beullens [10]. We first describe the attack against VDOO and then compute the required attack complexity to perform this attack against VDOO. Attacker starts with $n \times m$-rectangular matrices $M_1, M_2, \cdots, M_n$

over $\mathbb{F}_q$ where each M_i is defined as

$$M_i = \begin{bmatrix} \mathcal{DP}(\mathbf{s}_1,\mathbf{s}_i) \\ \mathcal{DP}(\mathbf{s}_2,\mathbf{s}_i) \\ \vdots \\ \mathcal{DP}(\mathbf{s}_n,\mathbf{s}_i) \end{bmatrix}$$

where $(\mathbf{s}_i)_{i=1}^n$ is a basis of $\mathbb{F}_q^n$.

Let $\mathbf{o}_2 \in \mathbb{F}_q^n$. The bi-linearity of $\mathcal{DP}$ implies

$$M := \sum_{i=1}^{n} o_{2i} M_i := \begin{bmatrix} \mathcal{DP}(\mathbf{s}_1,\mathbf{o}_2) \\ \mathcal{DP}(\mathbf{s}_2,\mathbf{o}_2) \\ \vdots \\ \mathcal{DP}(\mathbf{s}_n,\mathbf{o}_2) \end{bmatrix}.$$

Hence, the maximum rank of M is o_2, since $\mathbf{o}_2 \in O_2$. This observation provides attacker a min-rank instance to find o_{2i}'s in $\mathbb{F}_q$.

To enhance the performance of the simple attack, Beullens combined the rectangular min-rank attack with the simple attack [11]. Like earlier, the attacker fixes $\mathbf{v}$ to get a linear map $\mathcal{DP}_{\mathbf{v}}$. This helps to find $\mathbf{o}_2 \in O_2$ using $\mathcal{DP}_{\mathbf{v}}(\mathbf{o}_2) = 0$.

This system of linear equations reduces the number of matrices by m in the rectangular min-rank instance. Thus, the basis of $\mathrm{Ker}(\mathcal{DP}_{\mathbf{v}})$ is $\mathbf{b}_1,\cdots,\mathbf{b}_{n-m}$. Hence, the new min-rank instance has $n-m$ matrices $\widetilde{M_i}$, where

$$\widetilde{M_i} := \sum_{j=1}^{n} b_{ij} M_j := \begin{bmatrix} \mathcal{DP}(\mathbf{s}_1,\mathbf{b}_i) \\ \mathcal{DP}(\mathbf{s}_2,\mathbf{b}_i) \\ \vdots \\ \mathcal{DP}(\mathbf{s}_n,\mathbf{b}_i) \end{bmatrix}, \quad \text{for } i=1 \text{ to } n-m.$$

If $\mathbf{y}$ is a solution of the new min-rank problem having $n-m$ matrices then $\mathbf{o}_2 = \sum_{i=1}^{n-m} y_i \mathbf{b}_i$ is a solution of the old min-rank problem. Hence, the attack needs to be repeated approximately q times, until it finds $\mathbf{o}_2 \in \mathrm{Ker}(\mathcal{DP}_{\mathbf{x}}) \cap O_2 \neq \{0\}$.

Attack Complexity. The number of field multiplications required to perform this attack is

$$3 \cdot q \cdot (n-m-1)(o_2+1) \binom{n}{r}^2 \cdot \binom{n-m+b-3}{b}^3$$

where b is the operating degree for the algorithm [7].

Example for SL-1 parameters. The attacker needs to guess a good $\mathcal{DP}_{\mathbf{x}}$. After then the attacker gets a min-rank instance of 60 matrices which has 159 rows and 100 columns and the span of these matrices has a matrix of rank 36. Bardet et al.'s [7] algorithm provides an efficient way to solve this min-rank instance. This computation needs 2^{133}-field operations.

4.4 Kipnis-Shamir Attack on VDOO

The attacker targeting VDOO can employ a technique similar to the one devised by Kipnis and Shamir [40] to retrieve the subspace O_2. This approach effectively aids in the separation of oil and vinegar variables, ultimately leading to the recovery of the private key. The complexity of this attack can be roughly estimated as $O(o_2^4 \cdot q^{n-o_2-1})$ field multiplications. To expedite this assault, the attacker leverages Grover's algorithm, which serves to reduce the complexity to $O(o_2^4 \cdot q^{(n-o_2-1)/2})$.

Example for SL-1 parameters. Attacker needs to perform approximately 2^{348}-field operations in classical settings and 2^{174}-field operations in quantum computer.

4.5 Intersection Attack on VDOO

Beullens introduced the intersection attack [10], which effectively reduced the claimed security level of the Rainbow signature scheme by approximately 20 bits compared to the original design. In this attack, Beullens improved upon the Rainbow band separation attack [25] using the analysis proposed by Perlner [47]. The intersection attack helps to identify k-vectors simultaneously within the oil-space O_2 by solving a system of quadratic equations for a vector within the intersection $\cap_{i=1}^{k} L_i O_2$, where L_i's are invertible matrices. This attack performs well when the intersection is non-empty, which occurs when $n < \frac{2k-1}{k-1} o_2$. The computational cost of this attack involves solving a quadratic system with $\binom{k+1}{2}^2 o_2 - 2\binom{k}{2}$ equations in $k(no_2) - (2k-1)o_2$ variables.

However, in the case of VDOO where $n \geq 3o_2$, there is no guarantee that the subspace (for more details, see [10]) namely $L_i O_2 \cap L_j O_2$ will exist. Consequently, the attack becomes probabilistic for VDOO and will succeed with a probability of $\frac{1}{q^{(n-3o_2+1)}}$.

Example for SL-1 parameters. The complexity to break SL-1 parameters, attacker needs 2^{131}-field multiplications.

4.6 Quantum Attacks

The attacker can accelerate certain aspects of the classical attacks using a quantum computer. For MQ- or OV-based schemes the only quantum algorithm that can help in cryptanalysis is Grover's search [37]. This algorithm reduces the search space, thereby reducing the number of field multiplications by a factor of $q^{k/2}$. This specifically does not threaten the post-quantum security of our scheme [19].

4.7 Provable security: EUF-CMA Security

Our VDOO scheme, similar to UOV, Rainbow, and other UOV-based signature schemes, offers universal unforgeability [24]. Like these other schemes, we incorporate a salt in the signature generation process to demonstrate the EUF-CMA security of our scheme. We have followed the established methodology for this purpose, as seen in prior work such as [52,12]. Here, we have only provided an outline of the proof. The full proof can be done using similar strategies as Mayo [12], QR-UOV [34], PROV [29], etc. Our security proof relies on the well-understood hardness of the UOV problem. We begin by defining the UOV problem and then introduce the VDOO problem.

For security reasons, we recommend that each salt value should be used for no more than one signature. Consequently, we fix the salt length at 16 bytes, assuming up to 2^{64} signature generations within the system [19].

Definition 3 (UOV Problem). *Suppose* $\mathsf{UOV}_{(n,v,o,q)}$ *denotes a family of UOV public polynomial maps where* n *is the number variables,* $v+o$ *is number of equations and* q *is the size of the finite field, and* $\mathsf{MQ}_{(q,n,m)}$ *denotes a family of random quadratic systems with* n *unknowns and* m *constraints over* $\mathbb{F}_q$*. The UOV problem asks to distinguish* $\mathcal{P}$ *from* $\mathsf{UOV}_{(q,n,v,o,)}$ *and* $\mathsf{MQ}_{(q,n,m)}$*. Suppose* $\mathcal{A}_{\mathsf{UOV}}$ *be the adversary solves the distinguishing problem and it has a distinguishing advantage as:*

$$\mathsf{Adv}_{\mathsf{UOV}}(\mathcal{A}_{\mathsf{UOV}})=\left|Pr[\mathcal{A}_{\mathsf{UOV}}(\mathcal{P})=1 \mid \mathcal{P}\in\mathsf{MQ}]-Pr[\mathcal{A}_{\mathsf{UOV}}(\mathcal{P})=1 \mid \mathcal{P}\in\mathsf{UOV}]\right|$$

It is widely believed that there is no probabilistic polynomial-time adversary, including quantum adversaries, denoted as $\mathcal{A}$, that can efficiently solve the UOV problem.

Definition 4 (VDOO Problem). *Suppose* VDOO *be a family of VDOO public polynomial map. Now given a random* $\mathcal{P}\in\mathsf{VDOO}$ *and* $\mathbf{t}\in\mathbb{F}_q^m$ *VDOO problem asks to find* $\mathbf{s}$ *such that* $\mathcal{P}(\mathbf{s})=\mathbf{t}$*. If* $\mathcal{A}$ *is such an adversary to compute the inverse of the VDOO public map then the advantage of this computation is*

$$\mathsf{Adv}_{\mathsf{VDOO}}(\mathcal{A}_{\mathsf{VDOO}})=Pr[\mathcal{P}(\mathbf{s})=\mathbf{t} \mid \mathcal{P}\in\mathsf{VDOO},\ \mathcal{A}_{\mathsf{VDOO}}(\mathcal{P},\mathbf{t})=\mathbf{s}]$$

Now we are going to state our main theorem which establishes the EUF-CMA security of the VDOO. To understand the security notion, we refer to [12,52,42].

Theorem 1. *Suppose the adversary* $\mathcal{A}$ *runs in time* T *to solve the* EUF-CMA *game of VDOO in the random oracle model. This adversary makes* q_s *signing queries and* q_h *random oracle queries. Then there exists* $\mathcal{A}_{UOV}$ *and* $\mathcal{A}_{VDOO}$ *running in time* $T+O((q_s+q_t)\cdot poly(q,v,d,o_1,o_2))$ *with*

$$\begin{aligned}\mathsf{Adv}^{\text{EUF-CMA}}_{\mathsf{VDOO}}(\mathcal{A}) \leq\ & \mathsf{Adv}_{\mathsf{UOV}_{(q,v',o')}}(\mathcal{A}_{UOV})+q_h\cdot\mathsf{Adv}_{\mathsf{VDOO}_{(q,v,d,o_1,o_2)}}(\mathcal{A}_{VDOO})\\ &+(q_s+q_h)q_s\cdot 2^{-|salt|}+q^{-m}.\end{aligned}$$

Proof idea. Here, we informally sketch the proof. We can adopt the proof methodology used in Mayo (see theorem 6 from [12]). In the first step, we can establish a reduction from the EUF-CMA security of the VDOO signature scheme to EUF-KOA (Existential unforgeability against key-only attack) security by simulating the signing oracle. Note that, the adversary does not have access to the signing oracle in the EUF-KOA game. Once this reduction is established, we can easily show a reduction from the UOV problem and VDOO problem to the EUF-KOA security game in the second step. Like the security proof of Mayo [12], we can use the hybrid proof system to establish both reductions. This proof style has also been adopted by many state-of-the-art OV-based constructions [34,35,29,23]. Finally, we can combine both of these two steps to establish the above theorem.

5 Parameters and Performance

This section describes our chosen parameters based on the security analysis described in Section. 4. We assess the *practicality* of the VDOO signature scheme, which involves a finely tuned trade-off among computation time, security, and communication costs. For most multivariate schemes, computation time is dominated by either the Gaussian elimination (solving linear system [2]) or the Gröbner basis method (solving quadratic system [3]). Communication cost is proportional to signature size + public key size.

5.1 Parameter Selection

Table. 1, shows the signature, public-key, and private-key sizes of VDOO for different security levels as determined by the parameter tuple (q,v,d,o_1,o_2). We follow the NIST classification [19] to categorize the parameters. We consider the complexity of two primary attacks: the simple attack [11] (SA) and the rectangular min-rank attack [10] (RA). From the attacker's point of view, these two attacks exhibit the most optimistic complexity among all other known attacks. Here, the complexity represents the number of field multiplications required for their execution.

Security level	params (q,v,d,o_1,o_2)	**Sign size (B)**	**Priv key size (KB)**	**Pub key size (KB)**	**Attacks (SA, RA)**
SL-I	(16,60,30,34,36)	96	243	236	(134,138)
SL-III	(256,100,30,40,40	226	1056	2437	(207,191)
SL-V	(256,120,50,60,70)	316	3524	8127	(270,264)

Table 1. VDOO parameter set for different NIST prescribed security levels

5.2 Comparison with other post-quantum schemes

In response to the NIST's last [19] and the latest [20] standardization call multiple post-quantum signatures schemes have been proposed based on MQ problem or its derivatives. For our comparative analysis, we focus on schemes with small signature sizes and well-established hardness assumptions only in Table. 2. For fairness, we compare with the parameters which provide at least 128-bit of classical security [19]. For details about the parameters of a scheme and their role in security and key sizes we kindly request interested readers to the original publications.

Signature schemes	**Computational bottleneck**	**Signature size (B)**	**Public key size (KB)**
VDOO (16,40,30,34,36)	$\mathsf{GE}_{(16,34)}$, $\mathsf{GE}_{(16,36)}$	96	238

[2] $\mathsf{GE}_{(q,n)}$: Gaussian elimination on a linear system with n unknowns and n linear equation over $\mathbb{F}_q$. This computation needs $O(n^3)$-field operations.

[3] $\mathsf{XL}_{(q,n)}$: eXtended Linearization or Gröbner basis method to solve a quadratic system of n variables and n constraints over $\mathbb{F}_q$. This computation needs 2^{2^n}-field operations.

Table 2 continued from previous page

Rainbow [24,36] (256,148,80,48)	$\mathsf{GE}_{(256,32)}$, $\mathsf{GE}_{(256,48)}$	164	258
IPRainbow [17] (257,32,32,38,7)	$\mathsf{GE}_{(257,32)}$, $\mathsf{GE}_{(257,38)}$, $\mathsf{XL}_{(257,7)}$	120	342.784
Mayo [12] (16,66,65,7,11)	$\mathsf{GE}_{(16,65)}$	387	1
QR-UOV [34,35] (7,740,100,10)	$\mathsf{GE}_{(7,100)}$	331	20.657
PROV[29] (136,46,8)	$\mathsf{GE}_{(8,46)}$	160	68.326
TUOV [23] (160,64,32,16)	$\mathsf{GE}_{(16,64)}$	80	65.552
VOX [33] (251,8,9,6,6)	$\mathsf{XL}_{(251,6)}$	102	9.1
UOV [13] (256,160,64,16)	$\mathsf{GE}_{(256,64)}$	96	66.576

Table 2: Compare with other multivariate signature for security level one (at least 128-bit) [19]

In Table. 3, we compare VDOO with recently standardized Crystals Dilithium [26], Falcon [32], SPHINICS+ [4] and recently submitted some signature schemes (see [20]) which are not based on MQ problem.

Comparisons/ Algorithms	VDOO	Crystals Dilithium	Falcon	Sphincs+	FuLeeca	LESS
Signature size (B)	96	2420	666	7856	1100	8400
Public key size (B)	23813	1312	897	32	1318	13700
Comparisons/ Algorithms	SQISign	Hawk	ASCON-Sign	MIRA	MiRitH	RYDE
Signature size (B)	177	555	7856	7376	7661	7446
Public key size (B)	64	1024	32	84	129	86

Table 3: Comparisons with other signatures for NIST security level 1

From the above tables, it is evident that VDOO outperforms the majority of existing multivariate signature schemes. This superiority stems from the smaller number of variables involved in Gaussian eliminations in VDOO. Furthermore, the signature generation process in VDOO does not rely on the Gröbner basis technique, which further confirms its practicality. Further Table. 3 illustrates that VDOO has one of the smallest signature sizes with respect to other quantum-safe signature schemes.

6 Conclusion

We have introduced a post-quantum signature algorithm, leveraging well established cryptanalysis techniques to devise a parameter set for VDOO. In order to ensure a minimum of 128-bit security, our scheme achieves a compact 96-byte signature size, which outperforms numerous existing signature schemes. Nonetheless, it does grapple with a sizable public key size, a challenge that is prevalent in a significant number of multivariate signature schemes.

Our immediate future endeavors will be centered around further compressing the public key size within the VDOO scheme. Additionally, we intend to delve into the exploration of VDOO's security within the quantum random oracle model (QROM). Subsequently, our focus will shift towards realizing hardware implementations and assessing potential physical attacks against our scheme.

References

1. Agrawal, M., Saxena, N.: Automorphisms of finite rings and applications to complexity of problems. In: Annual Symposium on Theoretical Aspects of Computer Science. pp. 1–17. Springer (2005) 8
2. Agrawal, M., Saxena, N.: Equivalence of $\mathbb{F}$-algebras and cubic forms. In: Annual Symposium on Theoretical Aspects of Computer Science. pp. 115–126. Springer (2006) 8
3. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., Smith-Tone, D.: Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. Online. Accessed 26th June, 2023 (2022), `https://nvlpubs.nist.gov/nistpubs/ir/2022/NIST.IR.8413-upd1.pdf` 2
4. Aumasson, J.P., Bernstein, D.J., Beullens, W., Dobraunig, C., Eichlseder, M., Fluhrer, S., Gazdag, S.L., H ulsing, A., Kampanakis, P., Kölbl, S., Lange, T., Martin M. Lauridsen, F.M., Niederhagen, R., Rechberger, C., Rijneveld, J., Schwabe, P., Westerbaan, B.: Sphincs+ submission to the nist post-quantum project, v.3.1 (2018), `https://sphincs.org/data/sphincs+-r3.1-specification.pdf`, [Online; accessed 10-June-2023] 2, 3, 20
5. Baena, J., Briaud, P., Cabarcas, D., Perlner, R., Smith-Tone, D., Verbel, J.: Improving support-minors rank attacks: Applications to GeMSS and Rainbow. In: Annual International Cryptology Conference. pp. 376–405. Springer (2022) 2
6. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.P., Verbel, J.: Algebraic attacks for solving the rank decoding and min-rank problems without Gröbner basis (2020). Preprint available on https://arxiv. org/pdf/2002.08322. pdf **3**, 22–30 2, 7, 13
7. Bardet, M., Bros, M., Cabarcas, D., Gaborit, P., Perlner, R., Smith-Tone, D., Tillich, J.P., Verbel, J.: Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 507–536. Springer (2020) 2, 16
8. Bernstein, D.J., Chou, T., Lange, T., von Maurich, I., Misoczki, R., Niederhagen, R., Persichetti, E., Peters, C., Schwabe, P., Sendrier, N., et al.: Classic McEliece: Conservative Code-based Cryptography. NIST submissions (2017) 2
9. Bettale, L., Faugere, J.C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology **3**(3), 177–197 (2009) 13, 14

10. Beullens, W.: Improved cryptanalysis of UOV and Rainbow. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 348–373. Springer (2021) 2, 4, 5, 6, 7, 12, 13, 15, 17, 19
11. Beullens, W.: Breaking Rainbow takes a weekend on a laptop. Cryptology ePrint Archive (2022) 3, 4, 6, 7, 13, 16, 19
12. Beullens, W.: Mayo: practical post-quantum signatures from oil-and-vinegar maps. In: Selected Areas in Cryptography: 28th International Conference, Virtual Event, September 29–October 1, 2021, Revised Selected Papers. pp. 355–376. Springer (2022) 2, 5, 7, 10, 13, 17, 18, 20
13. Beullens, W., Chen, M.S., Ding, J., Gong, B., Kannwischer, M.J., Patarin, J., Peng, B.Y., Schmidt, D., Shih, C.J., Tao, C., Yang, B.Y.: UOV: Unbalanced Oil and Vinegar Algorithm Specifications and Supporting Documentation Version 1.0 (2018), `https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/UOV-spec-web.pdf`, [Online; accessed 5-September-2023] 5, 20
14. Billet, O., Gilbert, H.: Cryptanalysis of Rainbow. In: International Conference on Security and Cryptography for Networks. pp. 336–347. Springer (2006) 2
15. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM. Cryptology ePrint Archive, Report 2017/634 (2017), `https://ia.cr/2017/634` 2
16. Buss, J.F., Frandsen, G.S., Shallit, J.O.: The computational complexity of some problems of linear algebra. Journal of Computer and System Sciences **58**(3), 572–596 (1999) 7
17. Cartor, R., Cartor, M., Lewis, M., Smith-Tone, D.: IPRainbow. In: International Conference on Post-Quantum Cryptography. pp. 170–184. Springer (2022) 3, 7, 20
18. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology - EUROCRYPT 2023 - 42nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Lyon, France, April 23-27, 2023, Proceedings, Part V. Lecture Notes in Computer Science, vol. 14008, pp. 423–447. Springer (2023). https://doi.org/10.1007/978-3-031-30589-4_15, `https://doi.org/10.1007/978-3-031-30589-4_15` 2
19. Chen, L., Moody, D., Liu, Y.: NIST post-quantum cryptography standardization. Transition **800**, 131A (2017) 2, 3, 17, 18, 19, 20
20. Chen, L., Moody, D., Liu, Y.K.: Post-quantum cryptography: Digital signature schemes. round 1 additional signatures, `https://csrc.nist.gov/Projects/pqc-dig-sig/round-1-additional-signatures` 2, 5, 7, 19, 20
21. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 392–407. Springer (2000) 13
22. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies. In: Advances in Cryptology–ASIACRYPT 2020: 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7–11, 2020, Proceedings, Part I 26. pp. 64–93. Springer (2020) 2
23. Ding, J.: Tuov: Triangular unbalanced oil and vinegar (2023) 5, 7, 10, 18, 20
24. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: International conference on applied cryptography and network security. pp. 164–175. Springer (2005) 2, 3, 5, 6, 10, 14, 17, 20
25. Ding, J., Yang, B.Y., Chen, C.H.O., Chen, M.S., Cheng, C.M.: New differential-algebraic attacks and reparametrization of Rainbow. In: International Conference on Applied Cryptography and Network Security. pp. 242–257. Springer (2008) 2, 13, 17

26. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(1), 238–268 (Feb 2018). https://doi.org/10.13154/tches.v2018.i1.238-268, `https://tches.iacr.org/index.php/TCHES/article/view/839` 2, 3, 20
27. Faugere, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of pure and applied algebra **139**(1-3), 61–88 (1999) 2, 13
28. Faugere, J.C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation. pp. 75–83 (2002) 2, 13
29. Faugere, J.C., Fouque, P.A., Macario-Rat, G., Minaud, B., Patarin, J.: PROV: PRovable unbalanced Oil and Vinegar specification v1. 0–06/01/2023 5, 13, 17, 18, 20
30. Faugere, J.C., Levy-dit Vehel, F., Perret, L.: Cryptanalysis of Min-Rank. In: Annual International Cryptology Conference. pp. 280–296. Springer (2008) 7
31. Feo, L.D., Jao, D., Plût, J.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. J. Math. Cryptol. **8**(3), 209–247 (2014). https://doi.org/10.1515/jmc-2012-0015, `https://doi.org/10.1515/jmc-2012-0015` 2
32. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-fourier lattice-based compact signatures over ntru (2018), `https://falcon-sign.info/`, [Online; accessed 10-June-2023] 2, 3, 20
33. France, T.D., Faugère, J.C., Fouque, P.A., Goubin, L., Larrieu, R., Macario-Rat, G., Minaud, B.: Principal submitter: Jacques patarin 5, 20
34. Furue, H., Ikematsu, Y., Hoshino, F., Kiyomura, Y., Saito, T., Takagi, T.: Qr-uov (2023) 5, 7, 13, 17, 18, 20
35. Furue, H., Ikematsu, Y., Kiyomura, Y., Takagi, T.: A new variant of unbalanced oil and vinegar using quotient ring: QR-UOV. In: 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 27. pp. 187–217. Springer (2021) 18, 20
36. Groups, G.: Rainbow round3 official comment (2022) 2, 3, 20
37. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing. pp. 212–219 (1996) 17
38. Johnson, D.S., Garey, M.R.: Computers and Intractability: A Guide to the Theory of NP-completeness. WH Freeman (1979) 2, 5
39. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 206–222. Springer (1999) 2, 5, 10
40. Kipnis, A., Shamir, A.: Cryptanalysis of the Oil and Vinegar signature scheme. In: Annual international cryptology conference. pp. 257–266. Springer (1998) 2, 5, 8, 17
41. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Annual International Cryptology Conference. pp. 19–30. Springer (1999) 7
42. Kosuge, H., Xagawa, K.: Probabilistic hash-and-sign with retry in the quantum random oracle model. Cryptology ePrint Archive (2022) 18
43. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Workshop on the Theory and Application of Cryptographic Techniques. pp. 419–453. Springer (1988) 2
44. Miller, V.S.: Use of elliptic curves in cryptography. In: Conference on the theory and application of cryptographic techniques. pp. 417–426. Springer (1985) 1
45. Moh, T.: A public key system with signature and master key functions. Communications in Algebra **27**(5), 2207–2222 (1999) 2

46. Patarin, J.: The Oil and Vinegar signature scheme. In: Dagstuhl Workshop on Cryptography September 1997 (1997) 2, 5
47. Perlner, R., Smith-Tone, D.: Rainbow band separation is better than we thought. Cryptology ePrint Archive (2020) 17
48. Petzoldt, A., Bulygin, S., Buchmann, J.: CyclicRainbow–a multivariate signature scheme with a partially cyclic public key. In: International Conference on Cryptology in India. pp. 33–48. Springer (2010) 12
49. Petzoldt, A., Bulygin, S., Buchmann, J.: Selecting parameters for the Rainbow signature scheme. In: International Workshop on Post-Quantum Cryptography. pp. 218–240. Springer (2010) 12
50. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves. Quantum Inf. Comput. **3**(4), 317–344 (2003). https://doi.org/10.26421/QIC3.4-3, `https://doi.org/10.26421/QIC3.4-3` 1
51. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978) 1
52. Sakumoto, K., Shirai, T., Hiwatari, H.: On provable security of UOV and HFE signature schemes against chosen-message attack. In: International Workshop on Post-Quantum Cryptography. pp. 68–82. Springer (2011) 17, 18
53. Shamir, A.: Efficient signature schemes based on birational permutations. In: Annual International Cryptology Conference. pp. 1–12. Springer (1994) 2
54. Shor, P.W.: Algorithms for quantum computation: Discrete logarithms and factoring. In: Proceedings 35th annual Symposium on Foundations of Computer Science. pp. 124–134. Ieee (1994) 1
55. Smith-Tone, D., Perlner, R., et al.: Rainbow band separation is better than we thought (2020) 2, 13
56. Tao, C., Diene, A., Tang, S., Ding, J.: Simple matrix scheme for encryption. In: International Workshop on Post-Quantum Cryptography. pp. 231–242. Springer (2013) 2
57. Thomae, E.: A generalization of the rainbow band separation attack and its applications to multivariate schemes. Cryptology ePrint Archive (2012) 2, 13
58. Wiedemann, D.: Solving sparse linear equations over finite fields. IEEE transactions on information theory **32**(1), 54–62 (1986) 13
59. Wolf, C., Braeken, A., Preneel, B.: On the security of stepwise triangular systems. Designs, Codes and Cryptography **40**(3), 285–302 (2006) 15
60. Yang, B.Y., Chen, J.M.: Building secure tame-like multivariate public-key cryptosystems: The new TTS. In: Australasian Conference on Information Security and Privacy. pp. 518–531. Springer (2005) 2