VDOO: A small 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. Currently, 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 equations is one such quantum hard problem. Compared to lattice-based cryptography, cryptosystems based on multivariate equations have considerably smaller key sizes.

Unbalanced oil-vinegar is a classic 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. Our carefully chosen parameters can resist the recent attack proposed by Beullens which drastically reduces the security of Rainbow. 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 considerably smaller than lattice-based post-quantum signature schemes of similar security.

Keywords: Post-quantum · Digital signature · Multivariate Cryptography · Rainbow · 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 [41], elliptic-curve discrete logarithm or elliptic-curve Diffie-Hellman [34] 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 [44] and Proos-Zalka [40] respectively. Therefore, quantum-resistant hard problems have gained interest in designers for designing public-key cryptosystems for the future. National Institute of Standards and Technology initiated a procedure for the standardization of quantum-safe cryptographic primitives such as key encapsulation mechanisms (KEM), public-key encryption (PKE), and digital signature algorithms in 2016 [17].

In 2022, NIST finalized one KEM (Crystals-Kyber [13]) and three signature schemes (Sphincs+ [8], Crystals-Dilithium [21], and Falcon [29]) after rigorous scrutiny spanning multiple years. Among these only Sphincs+ is based on cryptographically secure hash functions, while Crystals-Kyber (KEM), Crystals-Dilithium, and Falcon rely on lattice-based hard problems. As the majority of these constructions are lattice-based, a breakthrough in the cryptanalysis of lattice-based cryptography can put the whole plan to migrate to post-quantum cryptography in jeopardy. Such incidents are not uncommon. Recently, Decru et al. [16] proposed an attack to completely break the security of supersingular isogeny Diffie-Hellman [25] 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) [36,30], isogeny-based [18], and code-based [7]. Standardizing cryptographic primitives necessitates a comprehensive investigation. NIST reissued a call for quantum-safe signature schemes to diversify the underlying difficulty (the deadline is in 2023). Due to its small signature size, multivariate oil-vinegar construction has garnered considerable attention during this standardization process.

Multivariate cryptography is a subfield of cryptography that relies on the intractability of root findings of multivariate quadratic (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 [29]. Numerous schemes, such as Matsumoto-Imai encryption scheme [33], Oil-Vinegar [36] signature, Rainbow [19] signature, Triangular schemes [35,43,49] signature, Simple Matrix encryption [46], and Mayo [11], have been developed as a result of multivariate cryptography. Patarin first proposed the Oil-Vinegar signature [36]. Kipnis and Shamir showed how to forge Patarin’s proposal [31]. Then Kipnis, Patarin, and Goubin upgraded the signature scheme by proposing Unbalanced Oil-Vinegar (UOV) [30].

Ding and Schmidt proposed Rainbow in 2005 [19]. The main interesting part of Rainbow is the multi-layer approach. The ground layer is the grand old scheme of unbalanced oil-vinegar [33]. Rainbow is comparatively more efficient than UOV due to its construction. This scheme was submitted to NIST-competition [17] and appeared as a third-round candidate. Due to the inclusion in the NIST competition, the cryptanalysis of Rainbow has been a well-studied area for the last decade. It includes direct attack [5,22,23], min-rank attack [12,5,14], bandseparation attack [20,47,45], rectangular min-rank and intersection attack [9]. In 2023, Beullens proposed a devastating cryptanalysis and discarded the security
level one (SL-1) parameter set of Rainbow. Rainbow team suggested using the old SL-3 parameter set as SL-1 parameters [27].

In 2022, Cartor et al. internally perturbed the second layer of Rainbow by mixing oil variables quadratically [15]. However, this mixing significantly increased the signature generation time. Also, parameter sets proposed by designers are not practical in terms of efficiency.

1.1 Our Contribution and Motivation

Cartor et al. modification increased signature generation time [15]. Also, increasing the parameter set for Rainbow increased the signature generation time, because the size of the linear system for Gaussian elimination will also increase. So our proposal is to reduce the computation time of Rainbow by using one diagonal layer instead of an oil layer.

We propose 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. Three-layer Rainbow needs three Gaussian elimination, however, our proposal needs two Gaussian elimination. We name our proposal as Vinegar-Diagonal-Oil-Oil (VDOO). The following structure illustrates the construction.

```
d-many diagonal
  Vinegar x_1, \ldots, x_v
  \vdots
  Vinegar x_1, \ldots, x_v
  Diagonal x_{v+1}

  \vdots

  Vinegar x_1, \ldots, x_v
  Diagonal x_{v+d}

o_1-many oil
  Vinegar x_1, \ldots, x_{v+d}
  \vdots
  Vinegar x_1, \ldots, x_{v+d}
  Oil x_{v+d}, \ldots, x_{v+d+o_2}

  \vdots

  Vinegar x_1, \ldots, x_{v+d}
  Oil x_{v+d}, \ldots, x_{v+d+o_1}

o_2-many oil
  Vinegar x_1, \ldots, x_{v+d+o_1}
  \vdots
  Vinegar x_1, \ldots, x_{v+d+o_1}
  Oil x_{v+d+o_1+1}, \ldots, x_{v+d+o_1+o_2}
```

**Remark:** To prevent the Beullens simple attack, the designer suggested increasing the parameter set to resist the attack. So if we are using Rainbow then the cost of the Gaussian elimination will be a bit high. Since the complexity of Gaussian elimination is \( N^3 \), therefore for Rainbow it is \( o_1^3 + o_2^3 \). Now for our scheme, the \( d \approx (o_1 + o_2)/3 \), \( o_1' = (o_1 + o_2)/3 \), and \( o_2' = (o_1 + o_2)/3 \). Therefore, the complexity for the Gaussian elimination is approximately \( o_1'^3 + o_2'^3 \). Thus this modification affects the performance significantly.
We analyze that simple attack can apply to our scheme also. To break our round-one parameter set, the simple attack requires $2^{134}$-field operations. Beullens combined the simple attack with the rectangular min-rank attack. This attack effectively reduces the security level of Rainbow. Like earlier, we perform the combined attack against our scheme. We find that the attack needs $2^{138}$ field operations to remove our level one parameter set.

Roadmap. In the upcoming Section 2 we present the construction of the traditional Rainbow and the subspace-oriented description of the Rainbow. Section 3 proposes a new post-quantum secure 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 compare our results with the state-of-the-art.

2 Prior results

Now we discuss the traditional description of Rainbow [19]. For simplicity, we always consider the homogeneous quadratic equations throughout the discussion. Rainbow uses three polynomial maps $S : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$, $T : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ and $F : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$. Among these maps, $S$ and $T$ are linear, and $F$ is a quadratic map which is known as a central map.

A tricky construction of the central map allows only the signer to invert the map efficiently. To explain the construction of the central map, we introduce some notations. We denote $[n]$ for the set $\{1, 2, \ldots, n\}$ and $[i : j]$ denotes $\{i, i+1, \ldots, j\}$. The construction uses an under-determined system that has $n$-variables and $m$-equations. According to designers, set of variables are splitted as $[n] = [v] \cup [v+1, v+o_1] \cup [v+o_1+1, v+o_2 = n]$, where $v, o_1, o_2$ are positive integers. Here first $v$-variables are vinegar variables and the rest are oil variables. In the first layer, the designer used $v$ variables vinegar and $o_1$ oil variables to construct $o_1$ homogeneous quadratic equations. So when one fixes values of vinegar variables then the quadratic system reduces to a linear system with $o_1$ variables and $o_1$ many constraints. Hence the Gaussian elimination can help to solve this linear system. Now in the second layer, all known $v + o_1$ variables are treated as vinegar variables and newly added $o_2$-many variables are oil variables. These variables help to construct $o_2$ many homogeneous quadratic equations. Therefore the mathematical expression for $l$-layer Rainbow is as follows.

$$f^{(k)}(x_1, x_2, \cdots, x_n) = \sum_{i,j \in [r]: i \leq j} \alpha_{ij}x_i x_j + \sum_{i \in [r]: j \in [r+1:r+o_2]} \beta_{ij}x_i x_j$$

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

KeyGen. Generate two random linear invertible affine transformation $S : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$ and $T : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n$. Construct the central map $F$. Therefore, the public polynomial map $P$ is $S \circ F \circ T$. Individual information of $S$, $F$, and $T$ are secret/private key and the composition map $P$ is the public key.
Signature. Algorithm takes secret key \((S, F, \text{ and } T)\), \(\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^n\) and message \(m\).

1. Compute \(t = \mathcal{H}(m)\)
2. Compute \(x = S^{-1}(t)\)
3. Construction of central map allows signer to compute \(y = F^{-1}(x)\)
4. Compute \(s = T^{-1}(t)\)

Output: Signature of the message \(m\) is \(s\)

Verification. Public polynomial map \(P\), signature \(s\), \(\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^n\) and message \(m\).

1. Compute \(t = \mathcal{H}(m)\)
2. \(t' = P(s)\)
3. If \(t = t'\) then output 1(success) else 0(failure)

2.1 Beullens Description

For a better view of cryptanalysis on Rainbow, Beullens explained the construction of Rainbow via subspaces \([9]\). Using this description Beullens derived a simple attack \([10]\). Before starting the discussion, we first define a differential polar form of a polynomial map.

The differential polar map of a polynomial map \(P\) is denoted by \(\mathcal{D}P : \mathbb{F}_q^n \times \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m\) and defined as

\[
\mathcal{D}P(x, y) = P(x + y) - P(x) - P(y).
\]

Now we first describe the trapdoor information of Rainbow. At first, the designer

chooses a secret chain of 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, one can construct a public polynomial map as follows.

- \(P\) maps each \(O_i\) to \(Q_i\) and its polar form satisfies (see Diagram)

\[
\text{for all } x \in \mathbb{F}_q^n \text{ and all } y \in O_i, \ P(x, y) \in Q_{i-1}.
\]

Fig. 1. \(l\) layer Rainbow
Now we discuss the inversion of $P$ using the knowledge of sequences of input and output subspaces. At first glance, for $l+1$-layer Rainbow, the value of the unknown $x$ can be represented as $v + o_1 + \cdots + o_l$ where all of the $o_i \in O_i$. Fix $v \in U \mathbb{F}_q^n$. Then $P$ is used in conjunction with the $i$-th layer’s output subspace $Q_i$ to calculate $o_i$. For the sake of clarity, let’s define the quotient space $O_i := \overline{O_i}/O_{i+1}$ now.

Using the knowledge of subspaces, the goal is to find $o_i$ for all $i$. This will lead to computing the preimage of any element from $\mathbb{F}_q^n$. For computing $o_i \in O_i$, use the following relation

$$
P(v + \overline{o_i}) + Q_i = y + Q_i$$

Earlier $v$ was 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 high probability. Repeatedly running this procedure, one can compute all $o_i$, which implies that preimage $x$ will be computed.

The signature scheme is the same as earlier. The secret key is sequences of subspaces instead of secret maps $S$, $T$, $F$. In a similar way, a verifier can verify the signature.

In 2022, Beullens discarded the level one parameter of Rainbow. He showed for small $n-m$, recovering all subspaces are significantly efficient. Also, small finite field sizes accelerate the attack. To protect against this attack Cartor et al. [15] proposed IPRainbow.

### 2.2 IPRainbow

Like Rainbow, IPRainbow is a multi-layer construction based on the ground layer UOV. The signing and verification phase is the same as Rainbow, the only difference is in the central polynomial. Central polynomials of the second layer are perturbed by $s$-many variables, which decreases the probability of guessing a variable in $O_2$ by $1/q^s$. However, running time is significantly increased during signature generation due to the presence of Gröbner basis technique in the inversion.

### 3 Our proposal: VDOO

Our scheme combines diagonals with oil-vinegar in an efficient manner. Suppose $[n]$ is the set of indices of variables. Pick the first $v$ variables as vinegar variables. Next $d$ variables are called diagonal variables. In this layer, we have $d$ many quadratic equations. In the $i$-th ($1 \leq i \leq d$) equation, only one variable appeared linearly and the rest $v + i - 1$-many variables appeared quadratically in the polynomial. In the next layer, we apply the oil-vinegar technique. This means we can generate $o_1$-many oil-vinegar polynomial using $v + d$-vinegar variables.
and newly added \( o_1 \)-oil variables. Like Rainbow, we again construct \( o_2 \)-many oil-vinegar polynomial using \( v + d + o_1 \)-vinegar variables and newly added \( o_1 + o_2 \)-oil variables. Finally, we have a quadratic system with \( n = v + d + o_1 + o_2 \) variables and \( m = d + o_1 + o_2 \)-many homogeneous quadratic equations.

### 3.1 Our Central Polynomials

Construction of central polynomial map \( F : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^m \) plays an important role in the multivariate signature schemes.

- **Diagonal Layer.** At first, we discuss the structure of the central polynomials for the diagonal layer. Suppose \( f_d(k) \) denotes a quadratic polynomial in this layer and \( k \in [v + 1 : v + d] \). The structure of the central polynomial is as follows.

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

- **First Oil Layer.** In this oil layer, we use \( v + d \) variables as vinegar variables and newly add \( o_1 \) variables as oil variables. All variables help us to construct \( o_1 \)-many central polynomials. Mathematical equations for first oil-layer central polynomials:

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

where \( k \in [v + d + 1 : \cdots, v + d + o_1] \) and \( \alpha_{ij}, \beta_{ij} \in \mathbb{F}_q \) (note: these field elements depend on \( k \) too).

- **Second Oil Layer.** This is top most oil layer. In this layer, we have \( v + d + o_1 \)-many vinegar variables and \( o_2 \) oil variables. In this layer, we have \( o_2 \) many quadratic equations. Those equations are of the form

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

### 3.2 Subspace Description of Central Polynomial

Our scheme can be explained through Beullens’s subspace descriptions. In this case, we have \( d + 2 \)-many input and output subspaces. These sequences are as follows.

- **Input subspaces**

\[
\mathbb{F}_q^n \supset O_{1,1} \supset O_{1,2} \supset \cdots \supset O_{1,d} \supset O_2 \supset O_3.
\]

- **Output subspaces**

\[
\mathbb{F}_q^m \supset Q_{1,1} \supset Q_{1,2} \supset \cdots \supset Q_{1,d} \supset Q_2 \supset Q_3 = \{0\}.
\]
From the diagram, these relations hold: \( \dim(O_{1,i}) = \dim(O_{1,i+1}) + 1 \) and \( \dim(Q_{1,i}) = \dim(Q_{1,i+1}) + 1 \) for \( 1 \leq i \leq d \). Also, \( \dim(O_{1,1}) = m \), \( \dim(O_{1,i}) = \dim(Q_{1,i-1}) \) for \( 1 < i \leq d \). In addition, \( \dim(O_2) = \dim(O_{1,d}) \), \( \dim(O_3) = \dim(Q_2) \).

![Diagram](image)

**Fig. 2.** Central polynomial of VDOO

Like Rainbow, the signer first fixes \( v \in U \mathbb{F}_q^n \). Since \( \dim(\tilde{O}_{1,i}) = \dim(O_{1,i}) - \dim(O_{1,i+1}) = 1 \), so for diagonal layer computing \( o_{1,1}, \ldots, o_{1,d} \) is very easy. Once these vector found, then update \( v \leftarrow v + o_{1,1} + \cdots + o_{1,d} \). Now like Rainbow, signer needs to solve for \( \tilde{o}_2 \in O_2(= O_2/O_3) \), so that

\[
\mathcal{P}(v) + \mathcal{P}(\tilde{o}_2) + D\mathcal{P}(v, o_2) = t \mod Q_2.
\]

We know, this equation is a linear system of \( o_2 \)-many variables and \( o_2 \)-many unknowns. With the overwhelming probability, the signer can able to compute \( o_2 \). Again update \( v \) and following a similar method, the signer finally computes the pre-image of \( t \). The main computation in the entire algorithm is the Gaussian elimination, whose complexity is \( O(N^3) \). Thus signing phase requires \( O(N^3) \)-time to compute the signature of a message.

### 3.3 VDOO Signature Scheme

**KeyGen.** This algorithm takes security parameters as input and output signer private key and public key.

1. For traditional view point or polynomial-map oriented observation, private key is the individual information three maps \( S : \mathbb{F}_q^n \to \mathbb{F}_q^n \), \( F : \mathbb{F}_q^n \to \mathbb{F}_q^n \) and \( T : \mathbb{F}_q^n \to \mathbb{F}_q^n \) (\( S \) and \( T \) are random invertible linear maps). Later compute \( \mathcal{P} = S \circ F \circ T \). The individual information of polynomial maps allows the private key holder to compute the inverse of \( \mathcal{P} \) efficiently.
2. For subspace description, secret key is the \(d\)-many diagonal vectors, oil subspace; that is input subspace sequence \(O_{1,1} \supset \cdots \supset O_{1,d} \supset O_2 \supset O_3\), and output subspace sequence \(Q_{1,1} \supset \cdots \supset Q_{1,d} \supset Q_2 \supset Q_3\). Using these subspace signer construct the public polynomial \(P\), so that \(P(O_3) = 0, P(x \in Q_2)\) when \(x \in O_2\), and \(P(x \in Q_{1,i})\) when \(x \in O_{1,i}\). Information about these subspaces is the secret key.

**Signature.** This algorithm takes the secret key, message, and a hash function \(\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^n\) as input.

1. Compute \(t = \mathcal{H}(m)\)
2. Compute the inverse of the public polynomial map \(P\), that is \(s = P^{-1}(t)\)
   1. For traditional view point Compute the inverse chain of mappings, that means, first compute \(S^{-1}\), then the inverse of the central map \(F^{-1}\) and later \(T^{-1}\)
   2. For subspace description Compute the inverse using the sequences of nested subspaces.
3. Publish \(s\) as a signature of the message \(m\).

**Verification.** This algorithm takes the public key, message, signature, and a hash function \(\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^n\) as input. It outputs either success or failure.

1. Compute \(t = \mathcal{H}(m)\)
2. Compute \(t' = P(s)\)
3. Check \(t = t'\), if holds output message is successfully verified otherwise report failure.

## 4 Security analysis of VDOO

The hardness of multivariate cryptography is based on three hard problems, multivariate quadratic problem, min-rank, and extended isomorphism problem (EIP). All cryptanalysis against multivariate schemes tries to find an optimal solution for small instances of these problems.

### 4.1 Hardness of Multivariate Cryptography

1. **MQ.** Knowing the public polynomial map \(P\) and \(y = P(x)\), the task is to find \(x\). This problem is called the MQ problem (multivariate quadratic), and it is known to be NP-hard [29].
2. **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

\[
\text{rank} \left( \sum_{i=1}^{k} m_i M_i \right) \leq r.
\]

This problem is called the min-rank problem and has proven to be NP-hard [11]. The min-rank problem appeared as a cryptanalytic tool in multivariate cryptography [32, 21, 10]. This attack helps to find a linear combination of public matrices which sums up to a low-rank matrix.
3. **EIP.** Find an equivalent composition of \( P = S' \circ F' \circ T' \), where \( S' \) and \( T' \) are equivalent affine maps, and \( F' \) 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 \([12]\)) but for some instances, polynomial time algorithms exist \([31]\).

4.2 Mathematical Cryptanalysis

In this discussion, we include all celebrated cryptanalysis results from popular schemes Rainbow and UOV. Because our scheme uses the construction of these schemes in a beautiful way. Due to the inclusion of Rainbow in the NIST competition, researchers focused on the cryptanalysis of Rainbow from the last decade. Many interesting results came out as output of the efforts, like direct attack \([5,22,23]\), min-rank attack \([12,5,6,4]\), band-separation attack \([20,47,45]\), rectangular min-rank and intersection attack \([9]\), and most famous simple attack \([10]\). Most of the attacks try to find a vector in the top oil layer, and then using this information the attacker is able to learn the entire subspace. Finding this vector either requires to use a direct attack or a min-rank algorithm.

- **Direct Attack.** This attack is the most fundamental attack for any multivariate scheme. These schemes adopted an under-determined system that has \( n \)-unknowns and \( m \)-homogeneous equation \((n > m)\). A hybrid approach converts this system to a determined system by fixing \( n - m \) variables. The time complexity of this attack, using \([8]\) approach is (in terms of field multiplications as):

\[
\min_{0 \leq k \leq m} q^k \cdot 3 \left( \binom{m - k + d}{d} \right)^2 \left( \binom{m - k}{2} \right)
\]

where \( k \) is the number of variables to be fixed during the algorithm and \( d \) is the smallest integer for which the coefficient of \( t^d \) in the series \((1 - t^2)^m/(1 - t)^{m-k}\) is non-positive. Quantum algorithms rely on Grover’s search \([28]\) to reduce the search space, that is the number of field multiplications is reduced by a factor of \( q^{k/2} \).

**Direct attack against VDOO.** Suppose level one parameter set has 160-variables and 100-constraints. According to \([5]\), 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 \(2^{280}\).

- **Beullens Simple Attack.** In 2022, Beullens proposed *simple attack* against Rainbow. This devastating attack reduces \( n \)-unknown and \( m \)-constraints in the quadratic system to \( n - m \)-unknown and \( m \)-constraints. At first, we present the basic idea of the simple attack in step by step manner.

**Input:** Public polynomial map \( P \)

**Output:** Recover sequences of subspaces.
**Step-I**: Choose \( x \in \mathbb{F}_q^n \) then solve for \( o \in O_2 \):
\[
DP_x(o) = 0 \\
\mathcal{P}(o) = 0
\]
Note that, \( DP_x \) indicates that \( x \) is fixed, so this is a linear system.

**Step-II**: Retrieve \( Q_1 \) using the information \( o \in O_2 \)
\[
\text{Span}\{DP_o(e_1), \ldots, DP_o(e_n)\} \subseteq Q_1
\]
for some linearly independent vectors. For enough such \( e_i \)'s equality will hold.

**Step-III**: To recover \( O_2 \), solve the following system of linear equations.
Because with high probability kernel matches with \( O_2 \).
\[
DP_o(e_1) \equiv 0 \mod Q_1 \\
DP_o(e_2) \equiv 0 \mod Q_1 \\
\vdots \\
DP_o(e_n) \equiv 0 \mod Q_1
\]

**Step-IV**: Once \( Q_1 \) and \( O_2 \) are retrieved then use Kipnis-Shamir attack to recover \( O_1 \).

**Complexity**: Complexity of Step-I dominates the complexity of other steps involved in this algorithm. Basically, a system of \( n \) variables and \( m \) non-linear equations reduce to a system of homogeneous equations with \( n-m \) variables. This computation can be performed via direct attack and it requires
\[
3 \cdot q \left( \frac{n-m-1+d}{d} \right)^2 \left( \frac{n-m-1}{2} \right)
\]
where \( d \) is the operating degree of the algorithm. That is, \( d \) 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.

**Simple Attack on VDOO**. As we mentioned, simple attack can help an attacker to find a vector in \( O_3 \) for our new scheme VDOO. As discussed earlier, the dominating complexity for the attack lies in the XL algorithm for solving a quadratic system of \( n-m \)-unknowns and \( m \)-constraints. In a step-by-step manner, we sketch the cryptanalysis of our scheme.

**Input**: Public polynomial map \( \mathcal{P} \).

**Output**: Recover sequences of subspaces.

**Find a vector \( o \in O_3 \)**: With probability \( 1/q \) (approximately), the attacker guesses a vector in \( O_3 \). To find this vector, an attacker should solve the following system.
\[
DP_x(o) = 0 \\
\mathcal{P}(o) = 0
\]
Like earlier, one can choose \( x \in U \mathbb{F}_q^n \). The attacker deploys the XL algorithm to solve the quadratic system of \( n - m \)-unknowns and \( m \)-constraints.

**Recovering \( Q_2 \) and \( O_3 \):** Attacker performs same steps to retrieve the \( Q_2 \) and \( O_3 \).

**Recovering vectors from \( O_1 \) and diagonal layer:** At this moment, the attacker is able to remove the top layer \( O_3 \). So (s)he reduces the earlier system to \( m_1 = m - o_3 \)-constraints in \( n_1 = n - o_3 \) variables (assume \( O_3 \) has \( o_3 \)-many variables). Attacker again applies the first step of this algorithm to find a vector in \( O_2 \), which means (s)he again needs to solve a quadratic system of \( n_1 - m_1 \)-unknowns and \( m_1 \) constraints. Once a vector in \( O_2 \) is found, then step-2 helps to recover the entire \( O_2 \) and \( Q_1 \). Therefore, the only task that remains to find is all diagonal vectors. Apply [10]'s trick to find all diagonal vectors in the layer. Here observe that the computation of finding a vector in \( O_3 \) dominates the computation of finding a vector in \( O_2 \).

**Example.** Suppose the security level one parameter set of our scheme VDOO is \( (n, q, v, d, o_1, o_2) = (160, 16, 60, 30, 34, 36) \). Apply Beullens trick to guess a vector in \( O_3 \), which happens with probability \( 1/q \). Finding one vector on \( O_3 \) asks to solve a quadratic system of \( 100 - \text{variables} \) \( 60 \)-unknowns. This computation is the most costly in the entire algorithm. Solving this quadratic system needs \( 2^{130} \) 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 143-bit security.

– **Rectangular Min-rank Attack.** Rectangular min-rank attack is proposed by Beullens [9]. Attacker starts with \( n \times m \)-rectangular matrices \( M_1, M_2, \ldots, M_n \) over \( \mathbb{F}_q \) where each \( M_i \) is defined as

\[
M_i = \begin{bmatrix}
\mathcal{DP}(s_1, s_i) \\
\mathcal{DP}(s_2, s_i) \\
\vdots \\
\mathcal{DP}(s_n, s_i)
\end{bmatrix}
\]

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

Let \( o_2 \in \mathbb{F}_q^n \). The bi-linearity of \( \mathcal{DP} \) implies

\[
M := \sum_{i=1}^n o_2 M_i := \begin{bmatrix}
\mathcal{DP}(s_1, o_2) \\
\mathcal{DP}(s_2, o_2) \\
\vdots \\
\mathcal{DP}(s_n, o_2)
\end{bmatrix}.
\]

Hence, the maximum rank of \( M \) is \( o_2 \), since \( o_2 \in O_2 \). This observation provides attacker a min-rank instance to find \( o_2 \)'s in \( \mathbb{F}_q \).

To enhance the performance of the simple attack, Beullens combined the rectangular min-rank attack with the simple attack [10]. Like earlier, the
attacker fixes $\mathbf{x}$ to get a linear map $D\mathcal{P}_x$. This helps to find $o_2$ via $D\mathcal{P}_x(o_2) = 0$.

This system of linear equations optimizes the number of matrices by $m$ in the rectangular min-rank instance. Thus, the basis of $\text{Ker}(D\mathcal{P}_x)$ is $b_1, \ldots, b_{n-m}$.

Hence, the new min-rank instance has $n - m$ matrices $\widetilde{M}_i$, where

$$\widetilde{M}_i := \sum_{j=1}^{n} b_{ij} M_j := \begin{bmatrix} D\mathcal{P}(s_1, b_i) \\ D\mathcal{P}(s_2, b_i) \\ \vdots \\ D\mathcal{P}(s_n, b_i) \end{bmatrix}, \text{ for } i = 1 \text{ to } n - m.$$

If $y$ is a solution of the new min-rank problem having $n - m$ matrices then $o_2 = \sum_{i=1}^{n-m} y_i b_i$ is a solution of the old min-rank problem. Hence, the attack needs to repeat approximately $q$ times, until it finds $o_2 \in \text{ker}(D\mathcal{P}_x) \cap O_2 \neq \{0\}$.

**Attack Complexity:** The number of field multiplications required to perform the 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 [6].

**Rectangular min-rank attack against VDOO.** This combined attack can be applied to our proposal also. Like simple attack, attacker expected a good guess for $D\mathcal{P}_x$. Then using this information, the attacker is able to get min-rank instances of 60 matrices. Each matrix has $n-1$ rows and $m$-columns and the span of these matrices has a matrix of rank $o_3$. Once a vector $o \in O_3$ is found, then it will be easy for an attacker to recover all vectors (as described in the simple attack against VDOO).

**Example** Our round parameters has 160 variables and 100 equations, and $q$ is 16. To break these parameters, the attacker needs to guess a good $D\mathcal{P}_x$. After then (s)he 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 [6] algorithm helps to the attacker to solve these min-rank instances. This computation needs $2^{133}$ field operations.

**Intersection Attack.** Beullens introduced the intersection attack [9]. This cryptanalysis reduced around 20 bits from what the designers of Rainbow claimed. In this attack, Beullens enhanced the Rainbow band separation attack [20] with the help of the analysis proposed by [37]. Like the simple attack, the attacker tried to find a vector in $O_2$ (for Rainbow) efficiently so that it satisfy the public polynomials.

**Intersection attack against VDOO.** Since this attack deals with finding a vector in the top layer. That is, for VDOO attacker tries to find a vector in $O_3$ using this variant of band-separation attack. We compute the complexity of this attack against our design. We get that $2^{131}$-field multiplications are required to discard our round one parameter set.
4.3 Provable security: EUF-CMA Security

Our VDOO, like Rainbow, simply provides universal unforgeability [19]. Since VDOO could be depicted as a multi-layer Rainbow utilizing UOV layers, we can make VDOO EUF-CMA secure with a change like [42]. [42] has detailed the proof for UOV. The same technique can be used to assert that the EUF-CMA security of the VDOO is the same as the security of the original Rainbow.

5 Parameter Selection of VDOO

Best on our cryptanalysis experience, we choose our parameters. At first, we fix parameters and then we compare our scheme with the state of the art.

5.1 Parameter Selection

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.

- Size of the central map $F$ for a triangular layer having depth $l$ is around

$$\sum_{i=1}^{l} \left( \frac{v_i(v_i + 1)}{2} + v_i \right)$$
field elements.

- Size of the central map $F$ for a UOV layer is around $o \times \left( \frac{v(v + 1)}{2} + ov \right)$
field elements (namely the $\alpha, \beta, \gamma, \delta$’s).

In the diagonal layer, the central polynomial $f_i$ has $v_i$-many are vinegar variables, and for the next central polynomial, $v_{i+1} = v_i + 1$. For the oil-vinegar layer $v$ and $o$ are the numbers of vinegar and oil variables in the UOV layer respectively. The size of the two affine transformations is as follows: for $S$ we need $m(m + 1)$, while for $T$ we need $n(n + 1)$, field elements. A smart designer can generate these random linear transformations from a 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 $m \frac{(n+1)(n+2)}{2}$. Just like the reductions used in Petzoldt et al. [39] and cyclicRainbow [38], we can also enhance the performance and significantly reduce the size of the public key in VDOO. It optimized the public key size from $O(mn^2 \log q)$ to $O(m^3 \log q)$.

We compute the key and signature sizes for VDOO for a wide range of parameter values in Table 1. This parameter set requires more careful study; which we leave as an open question. Here, we select parameters in accordance with NIST guidelines [17]. In this table, $n$ denotes the number of variables, $m$ number of equations, $q$ field size, $d$ is the number of diagonal variables, $o_2$-many oil variables present in $O_2$ and $o_3$-many oil variables present in $O_3$. In this table, we also put the complexity of the simple attack and the rectangular min-rank attack.
(especially the combined attack) [10]. Since both, of these attacks have the best complexity over other attacks, so we listed the complexity of these two attacks. Here complexities are given as the number of field multiplications required to perform the attack.

<table>
<thead>
<tr>
<th>SL</th>
<th>n</th>
<th>q</th>
<th>m</th>
<th>d</th>
<th>$\alpha_2$</th>
<th>$\alpha_3$</th>
<th>Private key size (KB)</th>
<th>Public key size (KB)</th>
<th>Public key size (KB) compressed</th>
<th>Signature size (byte)</th>
<th>simple attack</th>
<th>combined attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>160</td>
<td>16</td>
<td>100</td>
<td>30</td>
<td>36</td>
<td>243</td>
<td>644</td>
<td>238</td>
<td>96</td>
<td>134</td>
<td>138</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>210</td>
<td>256</td>
<td>110</td>
<td>30</td>
<td>40</td>
<td>1056</td>
<td>2437</td>
<td>875</td>
<td>226</td>
<td>207</td>
<td>191</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>300</td>
<td>256</td>
<td>180</td>
<td>50</td>
<td>60</td>
<td>3524</td>
<td>8127</td>
<td>2254</td>
<td>316</td>
<td>270</td>
<td>264</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Parameter set of VDOO

5.2 Comparison within similar security levels

In this section, we compare our proposal with Rainbow (with updated parameters [27,19]), IPRainbow [15], UOV, Mayo [11] (see Table 2).

<table>
<thead>
<tr>
<th>Signature Algorithm</th>
<th>Sign size (bit)</th>
<th>Private key size (KB)</th>
<th>Public key size (KB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDOO (16, 100, 30, 34, 36)</td>
<td>768</td>
<td>243</td>
<td>238</td>
</tr>
<tr>
<td>Rainbow [27] (SL1) (256, 148, 80, 48)</td>
<td>1312</td>
<td>611.3</td>
<td>258</td>
</tr>
<tr>
<td>UOV (256, 47, 71)</td>
<td>1072</td>
<td>276.9711</td>
<td>335.58</td>
</tr>
<tr>
<td>IPRainbow [15] (257, 32, 32, 38, 7)</td>
<td>944</td>
<td>220.320</td>
<td>342.784</td>
</tr>
</tbody>
</table>

Table 2. Comparison table for security level parameter set I

From the efficiency point of view, we can say that VDOO performs better than IPRainbow because the latter needs Gröbner basis algorithm in the signature phase. Gröbner basis algorithm is one of the most expensive algorithms; due to this heavy computation IPRainbow is approximately 50 times slower than our scheme.

Further, our diagonal trick reduces the size of Gaussian elimination in Rainbow. Hence, VDOO performs at least as well as both Rainbow and IPRainbow signature schemes. Mayo is one of the beautiful signature schemes designed in
2021. But it heavily suffers from large signature size. Considering all aspects, VDOO is a practical signature scheme.

References


27. Groups, G.: Rainbow round3 official comment (2022)


37. Perlner, R., Smith-Tone, D.: Rainbow band separation is better than we thought. Cryptology ePrint Archive (2020)


45. Smith-Tone, D., Perlner, R., et al.: Rainbow band separation is better than we thought (2020)


47. Thomae, E.: A generalization of the rainbow band separation attack and its applications to multivariate schemes. Cryptology ePrint Archive (2012)
