A New Multivariate Digital-Signature Scheme by Mixing Oil-Vinegar with Triangles

Anindya Ganguly and Nitin Saxena

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

Abstract. Multivariate cryptography is based on multivariate quadratic or MQ (i.e. multivariate quadratic root-finding) problem which is known to be NP-hard; thus it is conjectured to be post-quantum secure. UOV (unbalanced Oil-Vinegar) is a popular technique (Kipnis et.al.1999) used to design the central polynomials in an MQ-based signature scheme; for instance, the Rainbow scheme (ACNS'05) is one of the more well-known candidates. A powerful attack on Rainbow was recently proposed by Beullens (CRYPTO'22), which significantly decreases its security.
Our work proposes two new MQ-based digital-signature schemes, called *TriRainbow* (short for 'Triangular Rainbow'). TriRainbow effectively combines, in a simple way, two well-known methods— Rainbow and Triangular schemes. Our signature scheme, proposed in this work, is secure against Beullens's attack. TriRainbow needs one Gaussian elimination during the signing phase, so it is as efficient as Rainbow, in addition, it offers better security compared to Rainbow.

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

1 Introduction

Cryptography is an old mathematical art ensuring data security in our rapidly growing digital world. In 1970, Diffie and Hellman first introduced public key cryptography, and the corresponding hardness of the discrete logarithm problem (DLP) [19]. Later in 1977, Rivest, Shamir, and Adleman proposed a new public key cryptosystem based on the hardness of integer factorization problem (IFP) [43]. To reduce the public key size, Koblitz [32] and Miller [34] independently proposed elliptic curve cryptography, which was based on the hardness of elliptic curve discrete logarithm problem (ECDLP).

In 1994, Peter Shor proposed a quantum algorithm to solve IFP and DLP in polynomial time [46]. Thus, conventional cryptography is no more secure in the quantum era. However, trapdoors which are based on lattices (like SVP [27], LWE [42]) and multivariate quadratic [37,38] can prevent quantum attacks. So, NIST has called for standardization of post-quantum secure public key primitives [4]. In the last couple of years, researchers are developing quantum technology

rapidly [1]. Thus, many industries and government organizations have started working on post-quantum secure primitives to avoid quantum threats or attacks like *harvest now, decrypt later* [36].

The cryptography based on the hardness of multivariate quadratic (MQ) problem is called *multivariate cryptography. MQ problem* asks to solve a system of multivariate quadratic polynomials over $\mathbb{F}_q$. This is already known to be NP-hard [28]. Multivariate cryptography contributed many schemes like Matsumoto-Imai [33] encryption scheme, Hidden Field Equation (HFE) based cryptosystem [37,15], Oil-Vinegar [38] signature, Rainbow [20] signature, Triangular schemes [35,45,51] signature, Simple Matrix encryption [48], and many more.

Patarin proposed the first Oil-Vinegar signature scheme [38]. Later Kipnis and Shamir [30] showed how to forge the signature; and suitably updated the scheme to *Unbalanced* Oil-Vinegar (UOV) [29]. To enhance the performance and reduce the public key size, Ding and Schmidt proposed the Rainbow signature [20]. The construction of this scheme can be viewed as a multi-layer UOV signature scheme [29]. Rainbow was a third-round candidate in the NIST-PQC competition [5]. The cryptanalysis of Rainbow has been a well-studied area for the last decade. Cryptanalysis literature includes direct attack [7,22,23], min-rank attack [12,7,8,6], band-separation attack [21,49,47], rectangular min-rank and intersection attack [10]. In 2022, Beullens recovered the secret key of Rainbow (for the round one parameter set) within 53 hours on a laptop [11].

Later in 2022, Cartor *et al.* proposed another layer-based construction called IPRainbow [14]. They modified Rainbow using an internal perturbation by few quadratic monomials. However, their major drawback is that the inversion of the central map needs Gröbner basis computation, which increases its time complexity for signature generation.

1.1 Our Contribution and Motivation

We propose a new layer-based construction, which has one UOV layer and one triangular layer. A triangular layer means we are adding each new variable in the central polynomial one by one. The first variant of our proposal uses vinegar and oil variables in the first layer and triangular variables in the second layer. The second variant uses vinegar and triangular variables in the first layer and oil variables in the second layer.

Triangular Rainbow. The central polynomial map plays an essential role in the Rainbow construction. Beullens's *simple attack* revealed the subspaces due to the properties of the public polynomial and sequences of input and output subspaces. Here, we alter an Oil-Vinegar polynomial and a triangular polynomial in each layer. So we rename it *TriRainbow*. The motivation is to decrease the probability of guessing a vector in the input subspace. This modification allows us to use the old security level one (SL1) parameters, which were broken due to the simple attack. Further, we claim that our scheme remains efficient, as, the only computational bottleneck is Gaussian elimination. We have reduced the number of Gaussian elimination from two to one. Therefore, TriRainbow enjoys

better security and performance compared to Rainbow [25]. Here we consider layer-two TriRainbow. First, we elaborate on our version-one TriRainbow.

Version-One. Suppose the total number of variables is $n =: v_3$. The first v_1 variables are vinegar, the next $v_2 - v_1$ variables are oil, and the last $v_3 - v_2$ variables are called triangular variables. The first layer of $v_2 - v_1$ central polynomials are Oil-Vinegar, and the second layer of $v_3 - v_2$ central polynomials are triangular.

– First layer central polynomials:

$$f_1^{(k)}(x_1, x_2, \cdots, x_n) \;=\; \sum_{i=1}^{v_1}\sum_{j=1}^{v_1} \alpha_{ij} x_i x_j + \sum_{i=1}^{v_1}\sum_{j=v_1+1}^{v_2} \beta_{ij} x_i x_j + \sum_{i=1}^{v_2} \gamma_i x_i + \delta$$

where $k \in \{v_1+1, \cdots, v_2\}$ and α_{ij}, β_{ij}, γ_i, $\delta \in \mathbb{F}_q$ (note: these field elements depend on k too.).

– Second layer central polynomials:

$$f_2^{(k)}(x_1, x_2, \cdots, x_n) = x_k \cdot \lambda_k(x_1, x_2, \cdots, x_{k-1}) + \iota_k(x_1, x_2, \cdots, x_{k-1})$$

where λ_k is a linear and ι_k is a quadratic, and both are randomly chosen from $\mathbb{F}_q[x_1, x_2, \cdots, x_{k-1}]$, where $k \in \{v_2 + 1, \cdots, v_3\}$.

Version-Two. Here, the first v_1 variables are vinegar, the next $v_2 - v_1$ variables are triangular, and the last $v_3 - v_2$ variables are oil. Further, the first layer of $v_2 - v_1$ central polynomials are triangular and the second layer of $v_3 - v_2$ central polynomials are Oil-Vinegar.

– First layer central polynomials:

$$f_1^{(k)}(x_1, x_2, \cdots, x_n) = x_k \cdot \lambda_k(x_1, x_2, \cdots, x_{k-1}) + \iota_k(x_1, x_2, \cdots, x_{k-1})$$

where λ_k is a linear and ι_k is a quadratic, and both are randomly chosen from $\mathbb{F}_q[x_1, x_2, \cdots, x_{k-1}]$, where $k \in \{v_1 + 1, \cdots, v_2\}$.

– Second layer central polynomials:

$$f_2^{(k)}(x_1, x_2, \cdots, x_n) \;=\; \sum_{i=1}^{v_2}\sum_{j=1}^{v_2} \alpha_{ij} x_i x_j + \sum_{i=1}^{v_2}\sum_{j=v_2+1}^{n} \beta_{ij} x_i x_j + \sum_{i=1}^{n} \gamma_i x_i + \delta$$

where $k \in \{v_2+1, \cdots, v_3\}$ and α_{ij}, β_{ij}, γ_i, $\delta \in \mathbb{F}_q$ (note: these field elements depend on k too.).

Inversion of the UOV layer is already known, while the inversion of the triangular layer is equally easy due to the underlying triangular-shape linear-system. Like Rainbow, TriRainbow can be expressed using the differential polar form. This helps to better understand the cryptanalysis of the proposed scheme. Naively, TriRainbow can be visualized as an instance of Rainbow with layers $d + 1$, where d denotes the depth of the triangular layers. Therefore to recover

the secret key, the probability of guessing a vector in the input subspace is $1/q^d$. Most of the cryptanalytic methods known, try to find a vector in the input subspace [10,39,47]. Hence, the success probability of such attacks is naively $1/q^d$.

Organization of the paper In the upcoming section, we present the construction of the traditional Rainbow and the polar form description of the Rainbow. This section also describes the simple attack and the newly modified signature IPRainbow. Section 3 proposes a new post-quantum secure multivariate signature scheme called TriRainbow or Triangular Rainbow. The cryptanalysis of our scheme is presented in Section 4. Further, in Section 5, we compare our results with the existing one, and a concluding remark is kept in the last section.

2 Prior Works

In 2005, Ding and Schmidt [20] proposed a multi-layer signature scheme Rainbow which is one of the most popular digital signature schemes based on Multivariate Cryptography. It is built on the older signature scheme UOV [29]. The motivation for such layer-based construction was to resist the Kipnis-Shamir attack [30]. The simple attack is the most efficient algorithm to recover the secret key (partially) [11]. The rectangular min-rank attack can be combined with the simple attack to make significant improvements to cryptanalysis [10,11]. Later Cartor *et al.* perturbed the central polynomial map of Rainbow to resist such attacks [14]. However, due to this perturbation, the time complexity increased during the signing phase. In this section, we discussed some preliminaries required, like, trapdoors, Rainbow signature and its polar form explanation, simple and rectangular min-rank attack, and IPRainbow signature scheme.

2.1 Trapdoors

The central polynomial map of any multi-layer signature scheme based on UOV, $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ is sandwiched by two randomly chosen invertible affine maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$ from both sides, i.e. the *public polynomial map* $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$. Now these individual maps $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$ are *secret* keys and $\mathcal{P}$ is the public key. Multi-layer-based signature schemes used MQ, min-rank and extended isomorphism problems as trapdoors.

1. **MQ.** Knowing the public polynomial map $\mathcal{P}$ and $\mathbf{y} = \mathcal{P}(\mathbf{x})$, the task is to find $\mathbf{x}$. This problem is called the MQ problem (multivariate quadratic), and it is known to be NP-hard [28].
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

$$\operatorname{rank}\left(\sum_{i=1}^{k} m_i M_i\right) \leq r.$$

   This problem is called the *min-rank problem* and has proven to be NP-hard [13]. The min-rank problem appeared as a cryptanalytic tool in multivariate

cryptography [31,24,7,10]. 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 $\mathcal{P} = \mathcal{S}' \circ \mathcal{F}' \circ \mathcal{T}'$, 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 [2,3]), but for some instances, polynomial time algorithms exist [30].

2.2 Rainbow

Traditional description. The construction given by Ding and Schmidt is known as the traditional description of Rainbow [20]. This discussion includes the central polynomial map, trapdoors, and affine linear maps and later describes the signature generation and verification process.

Central polynomial map. Let n be the number of variables and $1 < v_1 < v_2 < \cdots < v_{l+1} = n$ be some integer parameters. The *vinegar set* V_i and *oil set* O_i are defined as: $V_i = \{1, 2, \cdots, v_i\}$ and $O_i = \{v_i + 1, v_i + 2, \cdots, v_{i+1}\}$ and the cardinalities of V_i and O_i be v_i and o_i respectively. From the construction, $o_i = v_{i+1} - v_i$, and $O_i = V_{i+1} - V_i$. The nested sequence of the vinegar set V_i is defined as follows

$$V_1 \subset V_2 \subset V_3 \cdots \subset V_{l+1} = \{1, 2, \cdots, v_1, \cdots, v_2, \cdots, v_3, \cdots, n\}.$$

Rainbow central map needs $m = n - v_1$ central polynomials $f^{(v_1+1)}, \cdots, f^{(n)} \in \mathbb{F}_q[x_1, \cdots, x_n]$. These are as follows, inspired by UOV, in the layer r:

$$f^{(k)}(x_1, x_2, \cdots, x_n) = \sum_{i,j \in V_r;\ i \leq j} \alpha_{ij} x_i x_j + \sum_{i \in V_r;\ j \in O_r} \beta_{ij} x_i x_j + \sum_{i \in V_r \cup O_r} \gamma_i x_i + \delta$$

where for each $k \in O_r$, elements $\alpha_{ij}, \beta_{ij}, \gamma_i$ and δ are taken from $\mathbb{F}_q$; and r denotes the layer .

In each layer r, first v_i's are vinegar and the next $v_{i+1} - v_i$'s are oil variables. In the next layer, newly added variables are considered oil variables, and old variables are vinegar variables. Finally, include all central polynomials from different layers to construct the central map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, where $\mathcal{F}(\mathbf{x}) = (f^{(v_1+1)}(\mathbf{x}), \cdots, f^{(n)}(\mathbf{x}))$.

Inversion. An efficient inversion of the central map implies better performance for the multivariate signature scheme. An l-th layer Rainbow requires l-many Gaussian elimination (GE). Here, the inversion algorithm starts with fixing the first v_1 vinegar variables by choosing random values and feeding those values into $f^{(v_1+1)}, f^{(v_1+2)}, \ldots, f^{(n)}$ polynomials. In the first layer, use the message, to get the o_1 linear constraints with $x_{v_1+1}, \cdots, x_{v_2}$ as unknown variables; this is an MQ instance. Now, solve this linear system using GE and put the solution into $f^{(v_2+1)}, \cdots, f^{(n)}$. In the second layer, the first v_2 variables play the role of vinegar variables. Again, a system of o_2 linear equations with o_2 unknowns can be obtained. To compute the unknown $x_{v_2+1}, \cdots, x_{v_3}$ again use

the GE algorithm. Therefore, performing these steps repeatedly, $x_1, x_2, \cdots, x_n$ can be computed; which gives a solution of MQ. It may happen that inversion reports failure. To overcome such circumstances, (randomly) change the first v_1 vinegar-variable values and repeat all the steps.

Signature. Let $\mathbf{h} \in \mathbb{F}^m$ be the hash values of the arbitrary length message. The signer uses her/his knowledge of individual maps $\mathcal{S}, \mathcal{F}$, and $\mathcal{T}$ to sign the message. Recursively (s)he computes $\mathbf{w} = \mathcal{S}^{-1}(\mathbf{h})$, $\mathbf{x} = \mathcal{F}^{-1}(\mathbf{w})$ and $\mathbf{y} = \mathcal{T}^{-1}(\mathbf{x})$. Since $\mathcal{S}$ and $\mathcal{T}$ are invertible affine maps, so $\mathcal{S}^{-1}$ and $\mathcal{T}^{-1}$ will exist (and are computable by GE), and $\mathcal{F}^{-1}$ will follow the inversion operation described above. Finally, $\mathbf{y}$ is the *signature* of the document $\mathbf{h}$.

Verification. Verifier matches the hash value of $\mathbf{h}$ with $\mathbf{h}' = \mathcal{P}(\mathbf{y}')$. If $\mathbf{h}' = \mathbf{h}$ holds, then (s)he reports *signature is accepted*, otherwise *reject*.

This construction can be visualized using the polar form of the *public* polynomial $\mathcal{P}(\mathbf{y}) = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}(\mathbf{y})$. For that purpose, the following discussion introduces the polar form and later describes the structure of the Rainbow using the polar form.

Polar Form Description. Beullens first explained Rainbow using the polar form [10]. The homogeneous part of any multivariate quadratic polynomial can be expressed as a matrix. Suppose $p(\mathbf{x})$ is a multivariate quadratic polynomial, then $dp(\mathbf{x}, \mathbf{y})$ is the *differential polar form* of $p(\mathbf{x})$ and it is defined as

$$dp(\mathbf{x}, \mathbf{y}) \;:=\; p(\mathbf{x} + \mathbf{y}) - p(\mathbf{x}) - p(\mathbf{y}) + p(\mathbf{0})\,.$$

Hence, polar form $\mathcal{DP}$ of any multivariate quadratic map $\mathcal{P}$ is defined as

$$\begin{aligned}\mathcal{DP}(\mathbf{x}, \mathbf{y}) &:= (dp_1(\mathbf{x}, \mathbf{y}),\ dp_2(\mathbf{x}, \mathbf{y}),\ \cdots, dp_m(\mathbf{x}, \mathbf{y})) \\ &= (p_1(\mathbf{x} + \mathbf{y}) - p_1(\mathbf{x}) - p_1(\mathbf{y}) + p_1(\mathbf{0}), \cdots, \\ &\qquad\qquad p_m(\mathbf{x} + \mathbf{y}) - p_m(\mathbf{x}) - p_m(\mathbf{y}) + p_m(\mathbf{0})) \\ &=: \mathcal{P}(\mathbf{x} + \mathbf{y}) - \mathcal{P}(\mathbf{x}) - \mathcal{P}(\mathbf{y}) + \mathcal{P}(\mathbf{0})\,.\end{aligned}$$

Ingredients. To illustrate the polar form explanation of the l layer Rainbow, we need,

- Two sequences of nested subspaces:

$$O_l \subset O_{l-1} \subset \cdots \subset O_1 \subset O_0 = \mathbb{F}_q^n \text{ called } \textit{input subspaces} \text{ and}$$

$$Q_l \subset Q_{l-1} \subset \cdots \subset Q_1 \subset Q_0 = \mathbb{F}_q^m \text{ called } \textit{output subspaces}.$$

- A multivariate quadratic polynomial map $\mathcal{P}$ which maps each O_i to Q_i and its polar form satisfies the following condition (see Diagram 1)

$$\text{for all } \mathbf{x} \in \mathbb{F}_q^n \text{ and all } \mathbf{y} \in O_i\ ,\ \mathcal{DP}(\mathbf{x}, \mathbf{y}) \in Q_{i-1}\,.$$

- The *secret* information comprises these two sequences of nested subspaces; while the *hardness* is to compute $\mathbf{x}$ from a given *public* quadratic map $\mathcal{P}$ and message/hash $\mathbf{y}$, such that $\mathbf{y} = \mathcal{P}(\mathbf{x})$.

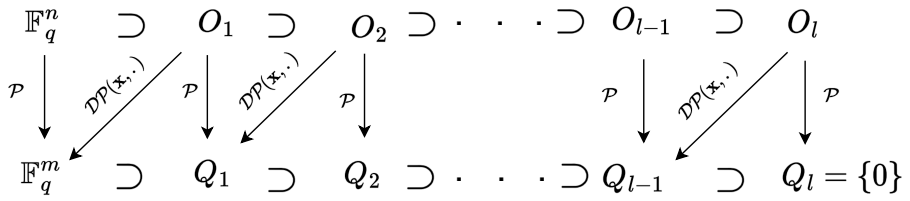

Fig. 1. l layer Rainbow

Inversion. This algorithm uses the secret information to compute $\mathbf{x} = \mathcal{P}^{-1}(\mathbf{y})$. Initially, the unknown $\mathbf{x}$ can be visualised as $\mathbf{v}+\mathbf{o_1}+\cdots+\mathbf{o}_l$, where each $\mathbf{o_i} \in O_i$. The first $\mathbf{v}$ is chosen randomly from $\mathbb{F}_q^n$; then using $\mathcal{P}$ and output subspace Q_i, we compute $\mathbf{o}_i$ from ith-layer. Let us introduce the *quotient space* $\overline{O}_i := O_i/O_{i+1}$ which will be useful in the later discussions.

At first layer: Let $\overline{\mathbf{o}}_1 \in \overline{O}_1$ be the first unknown that we want to find by solving the following equation mod Q_1:

$$\begin{aligned} &\mathcal{P}(\mathbf{v}+\overline{\mathbf{o}}_1) + Q_1 = \mathbf{y} + Q_1 \\ \Longrightarrow\ &\mathcal{P}(\mathbf{v}) + \mathcal{P}(\overline{\mathbf{o}}_1) + \mathcal{DP}(\mathbf{v}, \overline{\mathbf{o}}_1) + Q_1 \ = \ \mathbf{y} + Q_1\,. \end{aligned}$$

Since $\mathbf{v}$ is chosen randomly, so $\mathcal{P}(\mathbf{v})$ is constant and $\mathcal{DP}(\mathbf{v}, \overline{\mathbf{o}}_1)$ is linear in $\overline{\mathbf{o}}_1$. By the construction (see Figure 1), $\mathcal{P}(\overline{\mathbf{o}}_1) \in Q_1$. Hence, the above equation is a linear system over the quotient vector space $\mathbb{F}_q^n/Q_1$. This gives $\dim(\mathbb{F}_q^m/Q_1) = m - \dim(Q_1)$ many linear constraints. While the number of unknowns (to specify $\overline{\mathbf{o}}_1 \in \overline{O}_1$) is $\dim(\overline{O}_1) = \dim O_1 - \dim O_2 = m - \dim Q_1$. Hence, there exists a unique solution $\overline{\mathbf{o}}_1$ with probability $(1 - 1/q)$, which is quite high.

At second layer: Right now we know $\mathbf{v}+\overline{\mathbf{o}}_1$. Similarly, we solve for $\overline{\mathbf{o}}_2 \in \overline{O}_2$ using the following relation mod Q_2:

$$\mathcal{P}(\mathbf{v}+\overline{\mathbf{o}}_1+\overline{\mathbf{o}}_2) + Q_2 \ = \ \mathbf{y} + Q_2\,.$$

At last layer: From Figure 1, $Q_l = \{0\}$. Suppose $\mathbf{o}_l \in O_l$, now use the relation,

$$\begin{aligned} &\mathcal{P}(\mathbf{v}+\mathbf{o}_1+\cdots+\mathbf{o}_l) \ = \ \mathbf{y} \\ \Longrightarrow\ &\mathcal{P}(\mathbf{v}+\mathbf{o}_1+\cdots+\mathbf{o}_{l-1}) \ + \ \mathcal{P}(\mathbf{o}_l) \ + \ \mathcal{DP}(\mathbf{v}+\mathbf{o}_1+\cdots+\mathbf{o}_{l-1},\ \mathbf{o}_l) \ = \ \mathbf{y} \end{aligned}$$

In the LHS, the first term is constant because all $\mathbf{o}_i$'s are known till the $l-1$ layer. By definition, $\mathcal{P}(\mathbf{o}_l) = 0$. Thus, the above equation forms a linear system that has $\dim(O_l)$ unknowns and $\dim(O_l)$ constraints. Hence, a unique solution exists for $\mathbf{o}_l$ with probability $(1 - 1/q)$. There may be a failure, in which case randomly change $\mathbf{v}$ and repeat the entire algorithm.

This viewpoint of inversion algorithm, helped Beullens to explain the simple attack. He exploited the properties of input/output subspaces and the public polynomial maps.

2.3 Beullens Simple Attack

Since the NIST security level of Rainbow has only two layers, so his cryptanalysis considers two layers of Rainbow. Beullens attack consists of four steps:

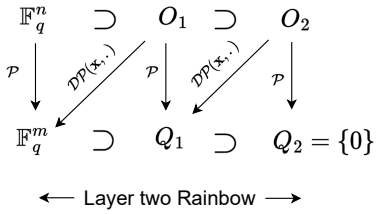

Fig. 2. Polar form description of Rainbow

1. **Find a vector $\mathbf{o}_2 \in O_2$:** At first, we fix $\mathbf{s} \in \mathbb{F}_q^n$ to get the linear map $\mathbf{t} \mapsto \mathcal{DP}(\mathbf{s}, \mathbf{t})$. From the property of polar form for fixed $\mathbf{s} \in \mathbb{F}_q^n$ and any $\mathbf{o}_2 \in O_2 \implies \mathcal{DP}(\mathbf{s}, \mathbf{o}_2) \in Q_1$. We call this linear map $\mathcal{DP}_\mathbf{s}$; it sends O_2 to Q_1. Also, $\dim Q_1 = \dim O_2 = o_2$. Hence, the kernel of the linear map $\mathcal{DP}_\mathbf{s}$ non-trivially intersects O_2 with a probability around $1/q$. So, the idea is to find a solution to the following system:

$$\begin{aligned} \mathcal{DP}_\mathbf{s}(\mathbf{o}_2) &= 0 \\ \mathcal{P}(\mathbf{o}_2) &= 0\,. \end{aligned}$$

   Here attacker gets a system of m homogeneous linear equations (from the first equation) and m homogeneous *quadratic* equations (from the second equation) with n unknowns of $\mathbf{o}_2$. Now m homogeneous linear equations reduce the number of unknowns. Hence, the task is to solve a system of m homogeneous *quadratic* equations with $n - m$ unknowns. If no solution is found, then randomly change $\mathbf{s}$ and re-execute these steps.

2. **Recover Q_1:** Randomly choose a basis $\{\mathbf{s}_i\}_i$ of $\mathbb{F}_q^n$ and compute $\mathcal{DP}(\mathbf{s}_i, \mathbf{o}_2)$, for $1 \le i \le n$. Then, with overwhelming probability,

$$\text{Span}\{\mathcal{DP}(\mathbf{s}_1, \mathbf{o}_2), \mathcal{DP}(\mathbf{s}_2, \mathbf{o}_2), \cdots, \mathcal{DP}(\mathbf{s}_n, \mathbf{o}_2)\} \;=\; Q_1 \text{ holds }\,,$$

   because, each $\mathcal{DP}(\mathbf{s}_i, \mathbf{o}_2) \in Q_1$.

3. **Recover O_2:** Once Q_1 is recovered, consider the linear system

$$\begin{aligned} \mathcal{DP}(\mathbf{s}_1, \mathbf{t}) &= 0 \mod Q_1 \\ \mathcal{DP}(\mathbf{s}_2, \mathbf{t}) &= 0 \mod Q_1 \\ &\vdots \\ \mathcal{DP}(\mathbf{s}_n, \mathbf{t}) &= 0 \mod Q_1\,. \end{aligned}$$

   Recall that for $\mathbf{o}_2 \in O_2$ we have $\mathcal{DP}(\mathbf{s}_n, \mathbf{o}_2) \in Q_1$. Therefore, the kernel equals O_2 with high probability.

4. **Recover O_1:** Once Q_1 and O_2 is found then the upper layer is erased. Thus, the problem reduces to a small parameter UOV instance $\mathcal{P}' : \mathbb{F}_q^{n-o_2} \longrightarrow \mathbb{F}_q^{m-o_2}$. Consequently, the Kipnis-Shamir attack [30] can be used to retrieve O_1.

Attack Complexity: Main computation is to solve $(m-1)$ random homogeneous quadratic equation in $(n-m-1)$ variables (for even characteristic field). This complexity dominates the computation of recovering O_1. Beullens used block Wiedemann XL algorithm [17,18] to solve the quadratic system. Therefore, the total number of field multiplications required for this attack is

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

where, d is the smallest positive integer for which the coefficient of t^d in the series $(1-t^2)^{m-1}/(1-t)^{n-m-1}$ is negative.

2.4 Rectangular Min-rank Attack

Beullens first designed the rectangular min-rank attack [10]. Let $M_1, M_2, \cdots, M_n$ be $n \times m$-rectangular matrices over $\mathbb{F}_q$ and 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 $(\mathbf{s}_i)_{i=1}^n$ forms a basis of $\mathbb{F}_q^n$.

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

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

Therefore, M has rank at most o_2 when $\mathbf{o}_2 \in O_2$; which gives us a min-rank instance to find o_{2i}'s in $\mathbb{F}_q$.

Beullens combined the rectangular min-rank attack with the simple attack [11]. He employed the linear map $\mathcal{DP}_{\mathbf{x}}$ for fixed $\mathbf{x}$. Earlier we have seen that $\mathcal{DP}_{\mathbf{x}}(\mathbf{o}_2) = 0$ is a useful linear constraint to find $\mathbf{o}_2$.

This system of linear equations helps to reduce the number of matrices by m in the rectangular min-rank instance. Thus, the basis of $\mathrm{Ker}(\mathcal{DP}_{\mathbf{x}})$ is $\mathbf{b}_1, \cdots, \mathbf{b}_{n-m}$. Therefore, 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 repeat approximately q times, until it finds $\mathbf{o}_2 \in \ker(\mathcal{DP}_{\mathbf{x}}) \cap O_2 \neq \{0\}$.

Attack Complexity: Beullens used Bardet *et. al* [8] algorithm for solving a min-rank instance with $n-m$ matrices of size $(n-1) \times m$. Note that, at this point, the attacker hopes that he successfully guessed a vector in O_2. Now the number of field multiplications required for 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 [8].

2.5 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 $1/q^s$. The first layer has OV central polynomial and any central polynomial in the second layer looks like:

$$f_{IPR}(x) = \sum_{i=1}^{v_2}\sum_{i=1}^{v_2} \alpha_{ij} x_i x_j + \sum_{i=1}^{v_2}\sum_{j=v_2+1}^{n} \beta_{ij} x_i x_j + \sum_{v_2+1}^{v_2+s}\sum_{j=v_2+1}^{v_2+s} \iota_{ij} x_i x_j + \sum_{i=1}^{n} \gamma_i x_i + \delta$$

where $\alpha_{ij}, \beta_{ij}, \iota_{ij}, \gamma_i, \delta \in \mathbb{F}_q$.

In particular, the second layer has $s \times s$-many oil$\times$oil monomials. So the next question is how to do the inverse. It is almost the same as Rainbow, but slower. After first layer computation, $x_1, x_2, \cdots, x_{v_2}$'s are known and $n - v_2$ variables are unknowns. Earlier, we have a linear system in oil variables. However, due to the perturbation, a "small" quadratic system of equations is present in $v_2+1, \cdots, v_2+s$ variables. So, we need to apply standard Gröbner basis technique to recover those values. Hence inversion adds an extra 2^{2^s} multiplier in the complexity. They showed that for sufficiently small s, Beullens simple attack succeeds with probability $1/q^{s+1}$. So, IPRainbow is quite expensive due to the presence of Gröbner basis technique in the inversion.

3 TriRainbow (Triangular Rainbow)

In this section, we present a new layer-based signature scheme that we call TriRainbow, or Triangular Rainbow; as it combines the benefits of both Rainbow and the Triangular signature schemes.

We are going to propose two versions of TriRainbow; where the first version has OV followed by a triangular polynomial and the second version has a triangular polynomial followed by an OV polynomial. Let us first introduce the new

central polynomial map, and then we will illustrate its inversion, the signature and the verification procedures.

Central map. First, we define the central polynomial map $\mathcal{F} : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^m$. We introduce a new set of variables called *triangular variables*. Similarly, define a *triangular set* which contains triangular variables. For these triangular variables, corresponding central polynomials are in a triangular fashion. For version one we first fix vinegar variables and then solve for oil variables. In the next layer, add some new triangular variables and treat known variables as vinegar variables.

For the second version, we swap oil variables with triangular variables. This means that in the first layer, fix vinegar variables and solve for triangular variables. In the next layer, all known variables are vinegar variables and newly added variables are oil variables.

3.1 Version One

For version one, every even layer has newly added variables as triangular variables; while for odd layers, newly added variables are oil variables. Therefore, central polynomials for even layers are triangular polynomials; while that for odd layers are OV polynomials.

- Layer one: $\underbrace{[1, 2, \cdots, v_1]}_{\text{vinegar}} \quad \underbrace{\{v_1 + 1, v_1 + 2, \cdots, v_2\}}_{\text{oil}}$
- Layer two: $\underbrace{[1, 2, \cdots, v_1, \cdots, v_2]}_{\text{vinegar}} \quad \underbrace{\{v_2 + 1, v_2 + 2, \cdots, v_3\}}_{\text{triangular}}$
- Layer three: $\underbrace{[1, 2, \cdots, v_1, \cdots, v_2, \cdots, v_3]}_{\text{vinegar}} \quad \underbrace{\{v_3 + 1, v_3 + 2, \cdots, v_4\}}_{\text{oil}}$
- Layer r (assuming odd): $\underbrace{[1, 2, \cdots, v_1, \cdots, v_2, \cdots, v_r]}_{\text{vinegar}} \quad \underbrace{\{v_r + 1, , \cdots, v_{r+1}\}}_{\text{oil}}$
- Layer r (assuming even): $\underbrace{[1, 2, \cdots, v_1, \cdots, v_2, \cdots, v_r]}_{\text{vinegar}} \quad \underbrace{\{v_r + 1, \cdots, v_{r+1}\}}_{\text{triangular}}$

For l-layer TriRainbow, V_r is a nested sequence of subsets (of n variables),

$$V_1 \subset V_2 \subset V_3 \cdots \subset V_{l+1} = \{1, 2, \cdots, v_1, \cdots, v_2, \cdots, v_3, \cdots, n\}.$$

Now define the vinegar set $V_r := \{1, 2, \cdots, v_r\}$, oil set $O_r := \{v_{2r-1} + 1, v_{2r-1} + 2, \cdots, v_{2r}\}$, and triangular set $T_r := \{v_{2r} + 1, v_{2r} + 2, \cdots, v_{2r+1}\}$. Also, define m-central polynomials $f^{(v_1+1)}, f^{(v_1+2)}, \cdots, f^{(n)} \in \mathbb{F}_q[x_1, \cdots, x_n]$.

- **Odd layer r:**

$$f_{odd}^{(k)}(x_1, \cdots, x_n) = \sum_{\substack{i,j \in V_r; \\ i \leq j}} \alpha_{ij} x_i x_j + \sum_{\substack{i \in V_r; \\ j \in O_{\frac{r+1}{2}}}} \beta_{ij} x_i x_j + \sum_{i \in V_r \cup O_{\frac{r+1}{2}}} \gamma_i x_i \ + \ \delta$$

  where, for each $k \in O_{\frac{r+1}{2}}$, pick α_{ij}, β_{ij}, γ_i, $\delta \in_U \mathbb{F}_q$.

- **Even layer r:**

$$f^{(k)}_{even}(x_1, \cdots, x_n) = \sum_{i\in[k-1]} \alpha'_i x_i x_k + \sum_{\substack{i,j\in[k-1];\\ i\leq j}} \beta'_{ij} x_i x_j + \sum_{i\in[k-1]} \gamma'_i x_i \ + \ \delta'$$

where for each $k \in T_{r/2}$, pick $\alpha'_i, \beta'_{ij}, \gamma'_i, \delta' \in_U \mathbb{F}_q$.

3.2 Version Two

Suppose $1, \cdots, v_1$ are vinegar variables and $v_1 + 1, \cdots, v_2$ are triangular variables in the first layer. Now randomly fix vinegar values. Then add one by one triangular variable to form $v_2 - v_1$-many central polynomials. This means any triangular polynomial in the first layer looks like:

$$f^{(k)}_1(x_1, \cdots, x_n) \ = \ x_k \cdot \lambda_k(x_1, x_2, \cdots, x_{k-1}) + \iota_k(x_1, x_2, \cdots, x_{k-1})$$

where λ_k is a linear and ι_k is a quadratic, and both are randomly chosen from $\mathbb{F}_q[x_1, x_2, \cdots, x_{k-1}]$, where $k \in \{v_1 + 1, \cdots, v_2\}$.

Now we generalize the definition for any layer r.

- Even layer r: $\underbrace{[1, \cdots, v_1, \cdots, v_2, \cdots, v_r]}_{\text{vinegar}} \ \underbrace{\{v_r + 1, , \cdots, v_{r+1}\}}_{\text{oil}}$.
- Odd layer r: $\underbrace{[1, \cdots, v_1, \cdots, v_2, \cdots, v_r]}_{\text{vinegar}} \ \underbrace{\{v_r + 1, \cdots, v_{r+1}\}}_{\text{triangular}}$.

This means that now the triangular set $T_r := \{v_{2r-1} + 1, v_{2r-1} + 2, \cdots, v_{2r}\}$, and oil set $O_r := \{v_{2r} + 1, v_{2r} + 2, \cdots, v_{2r+1}\}$. The central map $\mathcal{F}$ looks like:

- **Odd layer r:**

$$f^{(k)}_{odd}(x_1, \cdots, x_n) = \sum_{i\in[k-1]} \alpha_i x_i x_k + \sum_{\substack{i,j\in[k-1];\\ i\leq j}} \beta_{ij} x_i x_j + \sum_{i\in[k-1]} \gamma_i x_i \ + \ \delta$$

where for each $k \in T_{(r+1)/2}$, pick $\alpha_i, \beta_{ij}, \gamma_i, \delta \in_U \mathbb{F}_q$.

- **Even layer r:**

$$f^{(k)}_{even}(x_1, \cdots, x_n) = \sum_{\substack{i,j\in V_r;\\ i\leq j}} \alpha_{ij} x_i x_j + \sum_{\substack{i\in V_r;\\ j\in O_{\frac{r}{2}}}} \beta_{ij} x_i x_j + \sum_{i\in V_r\cup O_{\frac{r}{2}}} \gamma_i x_i \ + \ \delta$$

where for each $k \in O_{\frac{r}{2}}$, pick $\alpha'_{ij}, \beta'_{ij}, \gamma'_i, \delta' \in_U \mathbb{F}_q$.

Inversion. Inversion of the TriRainbow central map is very efficient. Two-layer TriRainbow has one UOV layer and one triangular layer. So far we know how to inverse UOV central polynomial map [29,20], and we adopt the same technique for the UOV layer. Inversion in the triangular layer is much simpler.

When one puts the vinegar values in the triangular polynomial then it reduces to a linear equation in x_k, that is

$$x_k \cdot \underbrace{h(x_1, \cdots, x_{k-1})}_{\text{constant}} = y_k - \underbrace{g(x_1, \cdots, x_{k-1})}_{\text{constant}}$$

where h and g are multivariate linear and quadratic polynomials respectively.

Do this process repeatedly until we recover all x_i's. The signature generation and verification processes are the same as Rainbow. Like Rainbow, TriRainbow needs two affine maps $\mathcal{S} : \mathbb{F}_q^m \longrightarrow \mathbb{F}_q^m$ and $\mathcal{T} : \mathbb{F}_q^n \longrightarrow \mathbb{F}_q^n$. The signature process uses knowledge of $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$ as *secret* information, and verification uses knowledge of $\mathcal{P}$ as *public* information.

TriRainbow can be explained using the *polar form* description. Using this description we will show that Beullens' attack can be adapted to TriRainbow albeit with a very low success probability. This allows the user to restore the SL1 parameter set. We will also analyse it via the known attacks related to UOV and Rainbow.

3.3 Polar Form Description

Both the variants of TriRainbow can be thought of as multi-layer Rainbow and each triangular polynomial is a UOV polynomial having a single oil variable. Hence it can be observed using the polar form description proposed by Beullens [10]. Since polar form description is the most friendly way for cryptanalysis, so we first present the polar form description of TriRainbow and then using this description we analyze the proposed signature scheme.

Version-One. TriRainbow version one has OV polynomials in the first layer and the second layer has triangular polynomials. First, we define the sequences of input and output subspaces. To avoid confusion, we denote O_{t_i} for input subspaces corresponding to each triangular layer, and Q_{t_i} for output subspaces corresponding to each triangular layer.

$$\text{Secret input subspaces: } \mathbb{F}_q^n \supset O_1 \supset \underbrace{O_{t_1} \supset \cdots \supset O_{t_l}}_{O_2} = \{*\}\,.$$

$$\text{Secret output subspaces: } \mathbb{F}_q^m \supset Q_1 \supset \underbrace{Q_{t_1} \supset \cdots \supset Q_{t_l}}_{Q_2} = \{0\}\,.$$

Note that the dimension of O_1 is m. The dimension of the subspace O_{t_1} is $m - o_1 = l = o_2$, while the dimension of each next subspace (in order) is one less than that of the previous subspace in the sequence, which means: $\dim(O_{t_2}) = m - o_1 - 1$, $\dim(O_{t_3}) = \dim(O_{t_2}) - 1$, etc. If the depth of the triangular layer is l, then $\dim(O_{t_l}) = 1 = \dim(Q_{t_{l-1}})$. Here, the depth denotes the number of triangular variables in the layer.

Version-Two. TriRainbow version two has vinegar variables and triangular variables in the first layer and the second layer has oil variables. So, the secret

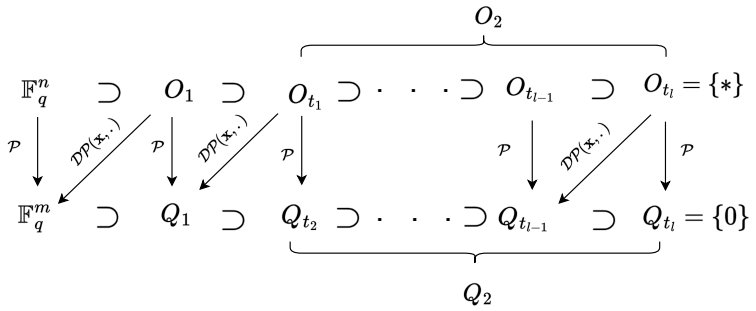

Fig. 3. Triangular Rainbow: Vinegar → Oil → Triangular

key is as follows:

$$\text{Secret input subspaces: } \mathbb{F}_q^n \supset \underbrace{O_{t_1} \supset \cdots \supset O_{t_l}}_{O_1} \supset O_2 \,.$$

$$\text{Secret output subspaces: } \mathbb{F}_q^m \supset \underbrace{Q_{t_1} \supset \cdots \supset Q_{t_l}}_{Q_1} \supset Q_2 = \{0\} \,.$$

Thus, $\dim(O_{t_1}) = m$, $\dim(O_{t_2}) = \dim(O_{t_1}) - 1$, etc. Like above, if the depth of the triangular layer is l, then $\dim(O_2) = o_2 = \#$ oil variables in the second layer $= m - l$.

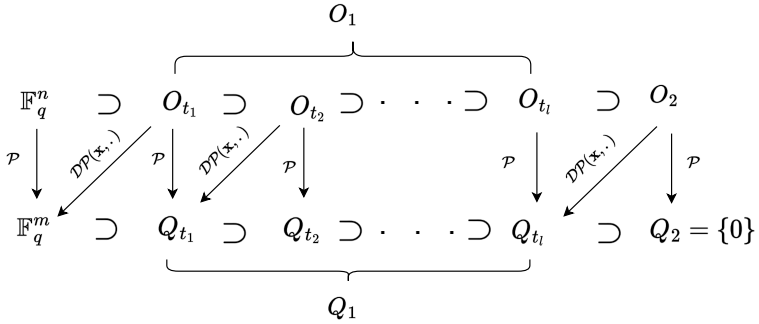

Fig. 4. Triangular Rainbow: Vinegar → Triangular → Oil

4 Cryptanalysis

At first, we analyze the simple attack, rectangular min-rank and combined attacks against TriRainbow. The intersection attack is an upgraded version of the Kipnis-Shamir attack. We apply it on newly proposed schemes. High-rank attack is important for schemes having triangular structures. Next, we discuss the complexity of the direct attack against proposed schemes. In the end, we apply the same trick to prove that both the versions of TriRainbow are EUF-CMA secure.

4.1 Simple Attack on TriRainbow

Now for cryptanalysis purposes, we first adopt Beullens simple attack. From Beullens attack, we have observed the following things.

- Attack starts with finding a vector $\mathbf{o}_2$ from O_2, and the probability of finding a vector $\mathbf{o}_2$ in O_2 is approximately $1/q$.
- When $\mathbf{o}_2$ is found then learn Q_1, and next learn O_2.
- Now, when the upper layer is removed, then it is a smaller parameter UOV.

Thus, for our two-layer TriRainbow, the probability of recovering all nested subspaces reduces multiplicatively to $1/q^l$. In our case, the attacker does not have the benefit of a small parameter UOV layer because of the depth l of the triangular layer. Similarly, the other attacks related to multivariate cryptography (OV based) also try to find a vector in O_2. We will present a brief discussion about the simple attack on TriRainbow (both versions).

Version-One. As we know, version one TriRainbow has ground layer UOV and the second layer is triangular. Therefore, for depth-l triangular layer, the dimension of the smallest input subspace is one, that is $\dim(O_{t_l}) = 1$ and $\dim(Q_{t_{l-1}}) = 1$.

We can apply Beullens strategy to find a vector $\mathbf{o}_{t_l}$ in O_{t_l} with probability $1/q$, and then recover $Q_{t_{l-1}}$ using the following relation

$$\langle \mathcal{DP}(\mathbf{o}_{t_l}, \mathbf{s}_1), \mathcal{DP}(\mathbf{o}_{t_l}, \mathbf{s}_2), \cdots, \mathcal{DP}(\mathbf{o}_{t_l}, \mathbf{s}_n) \rangle \subseteq Q_{t_{l-1}} ,$$

where $(\mathbf{s}_i)_{i=1}^n$ forms a basis of $\mathbb{F}_q^n$. Since $\dim(O_{t_l}) = 1$, so O_{t_l} is recovered. Thus, the upper layer is removed. Now doing a similar technique, the second upper layer can be removed with probability $1/q$, and thus the total probability is around $1/q^2$. Note that, this means that a partial key (for the upper layer) recovery is possible; to mitigate this issue we may add a few (based on the security level) dummy layers on the top.

As per NIST guidelines [16], SL1, SL3, and SL5 mean security levels are equivalent to the security level of AES-128, AES-192, and AES-256 respectively. If we adopt the Rainbow SL1 parameter set, $(q, n, m, o_2) = (16, 100, 64, 32)$, then the triangular layer's depth is 32. To avoid partial key recovery, we add two *dummy* layers on the top. This would modify the SL1 parameter set for TriRainbow to $(q, n, m, o_2) = (16, 102, 66, 34)$. Here o_2 is the depth of the triangular layer. Therefore, the probability of retrieving the entire key through a simple attack is approximately $1/q^{o_2}$ (which is very low).

If we adopt the Rainbow SL3 parameter set, $(q, n, m, o_2) = (256, 148, 80, 48)$, then 48 triangular layers are there. Like above, we can again add two more dummy triangular layers to prevent partial key recovery. Therefore, SL3 parameter set for TriRainbow becomes $(q, n, m, o_2) = (256, 150, 82, 50)$. Similarly, the SL5 parameter set is $(256, 197, 101, 65)$.

Version-Two. TriRainbow version two has vinegar and triangular variables in the first layer and oil variables in the second layer. Let the outer layer, O_2 has dimension o_2; therefore $\dim(Q_{t_l}) = o_2$. Now the goal is to find a vector $\mathbf{o}_2$ in

O_2, the simple attack can find such a vector with probability around $1/q$. Once $\mathbf{o}_2$ is found, then we use the following relation to recover Q_{t_l}.

$$\langle \mathcal{DP}(\mathbf{o}_2, \mathbf{s}_1), \mathcal{DP}(\mathbf{o}_2, \mathbf{s}_2), \cdots, \mathcal{DP}(\mathbf{o}_2, \mathbf{s}_n) \rangle \subseteq Q_{t_l} .$$

Once Q_{t_l} is recovered then again guess for $\mathbf{o}_{t_l} \in O_{t_l}$ with probability $1/q$; therefore total probability to remove the outer layer and second outer layer is $1/q^2$. Like version one TriRainbow, we may need to add a few dummy triangular layers (depending on the security level) to avoid partial key recovery through the simple attack. Note that, we cannot simply add one UOV layer because it increases complexity and key size. Hence using the simple attack, the total probability to retrieve all subspaces is around $1/q^l$. Parameter sets are the same as in version one.

4.2 Rectangular Min-rank Attack on TriRainbow

Any combination, of the simple attack with the rectangular min-rank attack, needs to guess a vector with probability $1/q$. Due to the depth-l of the triangular layer, the probability of guessing a vector will decrease exponentially with the depth.

4.3 Intersection Attack

Intersection attack is proposed by Beullens [10]. Basically, Beullens enhanced the Rainbow band separation attack [21] with the help of the analysis proposed by [39]. Like the simple attack, the attacker tried to find a vector in O_2 efficiently. Due to the triangular structure in our design, the attack complexity will increase exponentially with the depth of the triangular layer. Also, the attacker may be able to remove the topmost layer, but the dummy layer still protects the private key.

4.4 High-rank Attack

High-rank attack is a powerful attack against triangular schemes like STS, TPM [45,35,51]. The complexity of the high-rank attack reported in [50] is $O(mn^3Lq^r)$, where r is the depth of the layer, L is the number of total layers, m is the number of quadratic equations and n is the number of variables. For our parameter set one (based on SL1), $q = 2^4$, $L = 35$, $r = 32$, $n = 102$, and $m = 66$. Thus, the complexity for the high-rank attack is approximately 2^{159} field multiplications.

4.5 Direct Attack

So far we have seen that for multivariate signature schemes $n > m$ holds, where n is the number of unknowns and m is the number of homogeneous quadratic equations. To solve this via the hybrid approach of [9], it needs to convert to a determined system with m quadratic equations and m unknowns. So $n - m$

variables should be fixed. We can estimate the complexity of the direct attack, using [9], in terms of field multiplications as:

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

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 computers can use Grover's search algorithm [26] to reduce the search space, that is the number of field multiplications is reduced by a factor of $q^{k/2}$. In the following table, we have computed the attack complexity in terms of the number of field multiplications.

Algorithm	SL1	SL3	SL5
Quantum	126.5	203.5	244
Classical	166.5	235.5	286

Table 1. Complexity of direct attack in $\log_2(\#\text{steps})$; see Sec.4.1 for the parameters

4.6 EUF-CMA Security

Like Rainbow, our TriRainbow only offers universal unforgeability [20]. Since TriRainbow can be potentially visualized as a multi-layer Rainbow, using UOV layers, so a modification like [44] allows us to make both the versions of TriRainbow EUF-CMA secure. [44] has detailed the proof for UOV. The same technique can be adapted to state that EUF-CMA security of the modified TriRainbow is the same as the security of the standard Rainbow.

5 Key Size and NIST Parameter Selection

Let us compute the key size of our two-layer TriRainbow, which means that it has one Triangular and one UOV layer. First, we calculate the private key size and then the public key size.

- 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 (namely the $\alpha, \beta, \gamma, \delta$'s).
- Size of the central map $\mathcal{F}$ for a triangular layer having depth l is around

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

Note that, here v and o are the numbers of vinegar and oil variables in the UOV layer respectively. For the triangular layer, v_i's are vinegar variables at i-th depth

and $v_{i+1} = v_i + 1$. The size of the two affine transformations is as follows: for $\mathcal{S}$, we need $m(m+1)$ and for $\mathcal{T}$, we need $n(n+1)$ field elements.

Now we compute the size of the public key of standard TriRainbow. Recall that, the public polynomial map is defined as $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$. Each n-variate quadratic polynomial needs $\frac{(n+1)(n+2)}{2}$ field elements. Hence, the size of the public key is $m\frac{(n+1)(n+2)}{2}$. Just like the reductions used in Petzoldt *et al.* [41] and cyclicRainbow [40], we can also enhance the performance and significantly reduce the size of the public key in TriRainbow.

In Table 2, for various parameter sets, we calculate the size of the key and the signature for both versions of TriRainbow. Our signature and the public key sizes are the same for both the versions, while the private key sizes differ. So we maintain two columns for this purpose. However, this parameter set is tentative as it needs a more extensive cryptanalysis; which we leave as an open question. Here, we follow the NIST recommendation for parameter selection [16].

Security level	Parameters (q, n, m, o_2)	Sign size (bit)	Private key size (KB) Version: I	Private key size (KB) Version: II	Public key size	Security level
I	(16, 102, 66, 34)	536	92.753	100.689	176.748	145
III	(256, 150, 82, 50)	1328	546.754	612.226	941	207
V	(256, 197, 101, 65)	1704	1193.52	1361.918	1989.801	272

Table 2. Preferable parameter set TriRainbow for both versions

5.1 Comparison within similar security levels

In this section, we compare our layer-based construction, combining UOV and triangular, with other layer-based constructions based on UOV, like Rainbow [20] and IPRainbow [14] (see Table 3). From all such schemes, Rainbow was the NIST third-round finalist.

From the efficiency point of view, we can say that TriRainbow performs better than IPRainbow, because the latter needs Gröbner basis algorithm in the signature phase. Also, we know that Rainbow uses Gaussian elimination algorithms twice during the inversion of the central map, however, both versions of TriRainbow need only one Gaussian elimination. Hence, TriRainbow performs at least as well as both Rainbow and IPRainbow signature schemes.

6 Conclusion

So far, we have seen that the new proposal called TriRainbow, or triangular Rainbow, is as efficient as Rainbow. Its public key and signature sizes are smaller and it offers better security than Rainbow. We survey all existing attacks based on Rainbow and apply them to TriRainbow. We showed that the simple attack

Signature Algorithm	Sign size (bit)	Private key size (KB)	Public key size (KB)
TriRainbow-I (16, 102, 66, 34)	536	92.753	176.748
TriRainbow-II (16, 102, 66, 34)	536	100.689	176.748
Rainbow (SL1) (256, 148, 80, 48)	1312	611.3	861.4
UOV (256, 47, 71)	1072	276.9711	335.58
IPRainbow (257, 32, 32, 38, 7)	944	220.320	342.784

Table 3. Comparison table for security level parameter set I

is ineffective against TriRainbow. However, we did not explore whether it offers the same security level against fault and side-channel attacks. We leave further cryptanalysis, against TriRainbow, as future work. Meanwhile, it is expected that TriRainbow can replace Rainbow as a practical signature scheme.

Acknowledgements A.G. thanks Angshuman Karmakar (IIT Kanpur) for clarifying doubts and proofreading. N.S. thanks the funding support from DST-SERB (CRG/2020/000045) and N.Rama Rao Chair.

References

1. IBM quantum breaks the 100-qubit processor barrier. Blog (Nov 2021), https://research.ibm.com/blog/127-qubit-quantum-processor-eagle 2
2. 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) 5
3. 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) 5
4. Alagic, G., Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Liu, Y.K., Miller, C., Moody, D., Peralta, R., et al.: Status report on the first round of the NIST post-quantum cryptography standardization process. US Department of Commerce, National Institute of Standards and Technology (2019) 1
5. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Miller, C., Moody, D., Peralta, R., et al.: Status report on the third round of the NIST post-quantum cryptography standardization process. National Institute of Standards and Technology, Gaithersburg (2022) 2
6. 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

7. 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 minrank problems without Gröbner basis (2020). Preprint available on https://arxiv.org/pdf/2002.08322. pdf **3**, 22–30 2, 5
8. 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, 10
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) 16, 17
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, 9, 13, 16
11. Beullens, W.: Breaking Rainbow takes a weekend on a laptop. Cryptology ePrint Archive (2022) 2, 4, 9
12. Billet, O., Gilbert, H.: Cryptanalysis of Rainbow. In: International Conference on Security and Cryptography for Networks. pp. 336–347. Springer (2006) 2
13. 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) 4
14. Cartor, R., Cartor, M., Lewis, M., Smith-Tone, D.: IPRainbow. In: International Conference on Post-Quantum Cryptography. pp. 170–184. Springer (2022) 2, 4, 18
15. Casanova, A., Faugere, J.C., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J.: GeMSS: A Great Multivariate Short Signature. Ph.D. thesis, UPMC-Paris 6 Sorbonne Universités; INRIA Paris Research Centre, MAMBA Team ... (2017) 2
16. Chen, L., Moody, D., Liu, Y.: NIST post-quantum cryptography standardization. Transition **800**, 131A (2017) 15, 18
17. Cheng, C.M., Chou, T., Niederhagen, R., Yang, B.Y.: Solving quadratic equations with Xl on parallel architectures-extended version. Cryptology ePrint Archive (2016) 9
18. 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) 9
19. Diffie, W.: New direction in cryptography. IEEE Trans. Inform. Theory **22**, 472–492 (1976) 1
20. 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, 4, 5, 12, 17, 18
21. 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, 16
22. 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
23. 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

24. Faugere, J.C., Levy-dit Vehel, F., Perret, L.: Cryptanalysis of Min-Rank. In: Annual International Cryptology Conference. pp. 280–296. Springer (2008) 5
25. Groups, G.: Rainbow round3 official comment (2022) 3
26. 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
27. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: International Algorithmic Number Theory Symposium. pp. 267–288. Springer (1998) 1
28. Johnson, D.S., Garey, M.R.: Computers and Intractability: A Guide to the Theory of NP-completeness. WH Freeman (1979) 2, 4
29. 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, 4, 12
30. Kipnis, A., Shamir, A.: Cryptanalysis of the Oil and Vinegar signature scheme. In: Annual international cryptology conference. pp. 257–266. Springer (1998) 2, 4, 5, 9
31. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Annual International Cryptology Conference. pp. 19–30. Springer (1999) 5
32. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of computation **48**(177), 203–209 (1987) 1
33. 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
34. 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
35. Moh, T.: A public key system with signature and master key functions. Communications in Algebra **27**(5), 2207–2222 (1999) 2, 16
36. Ott, D., Peikert, C., et al.: Identifying research challenges in post-quantum cryptography migration and cryptographic agility. arXiv preprint arXiv:1909.07353 (2019) 2
37. Patarin, J.: Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 33–48. Springer (1996) 1, 2
38. Patarin, J.: The Oil and Vinegar signature scheme. In: Dagstuhl Workshop on Cryptography September, 1997 (1997) 1, 2
39. Perlner, R., Smith-Tone, D.: Rainbow band separation is better than we thought. Cryptology ePrint Archive (2020) 4, 16
40. 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) 18
41. 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) 18
42. Regev, O.: The learning with errors problem. Invited survey in CCC **7**(30), 11 (2010) 1
43. 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

44. 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
45. Shamir, A.: Efficient signature schemes based on birational permutations. In: Annual International Cryptology Conference. pp. 1–12. Springer (1994) 2, 16
46. 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
47. Smith-Tone, D., Perlner, R., et al.: Rainbow band separation is better than we thought (2020) 2, 4
48. 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
49. Thomae, E.: A generalization of the rainbow band separation attack and its applications to multivariate schemes. Cryptology ePrint Archive (2012) 2
50. Wolf, C., Braeken, A., Preneel, B.: On the security of stepwise triangular systems. Designs, Codes and Cryptography **40**(3), 285–302 (2006) 16
51. 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, 16