symmetry

MDPI

Article

New Identified Strategies to Forge Multivariate Signature Schemes

Nurul Amiera Sakinah Abdul Jamal [1,†], Muhammad Rezal Kamel Ariffin [1,2,*,†], Siti Hasana Sapar [2,*,†] and Kamilah Abdullah [1,3,†]

[1] Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[2] Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia
[3] Department of Mathematics, Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA Shah Alam, 40450 Shah Alam, Selangor, Malaysia
* Correspondence: rezal@upm.edu.my (M.R.K.A.); sitihas@upm.edu.my (S.H.S.); Tel.: +603-9769-6838 (M.R.K.A.)
† These authors contributed equally to this work.

Abstract: A rogue certificate authority (RCA) is a dishonest entity that has the trust of web browsers and users to produce valid key pairs which are vulnerable. This work analyses two acknowledged post-quantum secure Multivariate Quadratic Problem (MQP) based signature schemes, namely the UOV and Rainbow signature schemes that obtain their key pair from a potential RCA methodology. We revisit two and provide a novel RCA methodology that would enable adversaries to forge UOV and Rainbow signatures. We also lay out two strategies to identify whether the public parameters are generated by the first two methodologies. To this end, strategies to identify the third strategy remain elusive. As such, the UOV and Rainbow schemes remain vulnerable to forgery if it was forged via the third methodology.

Keywords: multivariate signature schemes; UOV; rainbow; Multivariate Quadratic Problem; rogue certificate authority; weak public key; post-quantum cryptography

check for updates

Citation: Abdul Jamal, N.A.S.; Kamel Ariffin, M.R.; Sapar, S.H.; Abdullah, K. New Identified Strategies to Forge Multivariate Signature Schemes. *Symmetry* **2022**, *14*, 2368. https://doi.org/10.3390/sym14112368

Academic Editors: Miodrag J. Mihaljevic and Chin-Ling Chen

Received: 30 September 2022
Accepted: 1 November 2022
Published: 10 November 2022

1. Introduction

Asymmetric key cryptosystems have solved the key distribution problem arising from the widespread use of symmetric key cryptosystems. In contrast to symmetric key cryptosystems which utilize the same key to encrypt and decrypt, asymmetric key cryptosystems use different keys known as public and private keys. Furthermore, asymmetric key cryptosystems not only solve the key distribution problem and provide confidentiality but also provide entity authentication, preserve message integrity and prevent identity repudiation. The public–private key pair is generated by the Certificate Authority (CA). However, a client might receive key pairs generated from a rogue Certificate Authority (RCA) who hides the fact that the produced key pairs have weaknesses that are only known to nobody else except the RCA [1]. The public key which works perfectly and satisfies the public security requirements during the key generation process, can be manipulated by an adversary in order to recover the secret parameters.

A digital signature is defined as a mathematical procedure which provides the authenticity and integrity of a message. The private signing key is used in the signing algorithm to sign the document and produce valid signatures; meanwhile, the public key is used in the verification algorithm to verify the validity of the signature corresponding to the document. In addition, forgery is an act of making a copy of a valid signature or document in order to deceive someone.

In 1994, [2] proved that classical cryptography will be no longer secure in the existence of a quantum computer. A quantum computer can solve hard problems such as Integer Factorization Problem (IFP) [3] and Discrete Log Problem (DLP) [4] in polynomial time. Hence, we require post-quantum cryptography algorithms which are resistant to a quantum computer. Among the candidates of post-quantum cryptography is multivariate cryptography.

In multivariate public key cryptography (MPKC), the underlying hard problem is defined as the Multivariate Quadratic Problem (MQP). In MQP, $\mathcal{P} = (p^{(1)}, \ldots, p^{(m)})$ is a system of m quadratic equations in n variables under the finite field $\mathbb{F}_q$. One needs to identify a vector $\mathbf{x} = (x_1, \ldots, x_n)$ such that the system of polynomials $\mathcal{P}(\mathbf{x}) = 0$ [5]. In order to forge a multivariate signature scheme, one has either to produce a valid signature $\mathbf{s}'$ such that $\mathcal{P}(\mathbf{s}') = \mathbf{z}' = \mathbf{z}$ or recover secret keys $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$.

The Unbalanced Oil and Vinegar (UOV) scheme by [6] requires one to choose o number of equations and $n = o + v$ number of variables where $v > o$. This is an amendment to the Oil and Vinegar (OV) scheme [7] that was successfully cryptoanalysed by Kipnis and Shamir attack [8]. The initial OV scheme sets $v = o$ [7]. On the other hand, the Rainbow signature scheme [9] is a multilayer version of UOV with smaller key and signature sizes which initiate better performance. In 2017, another version of UOV which utilizes smaller key and signature sizes was proposed and coined the LUOV cryptosystem [10]. In 2020, Petzolt proposed an algorithm to speed up the key generation of Rainbow [11].

The National Institute of Standards and Technology (NIST) announced the request for nominations for public key post-quantum cryptographic algorithms in 2016 in preparation for the quantum computing era. Since then, many quantum algorithms resistant to quantum computers have been proposed, including [12–16]. The Rainbow signature scheme successfully advanced from Round 1 to Round 2 and was one of the finalists for the third-round candidates for digital signature algorithms other than CRYSTALS-DILITHIUM [17] and FALCON [18] whereas, Classic McEliece [19], CRYSTALS-Kyber [20], NTRU [21] and SABER [22] are the finalists for encryption algorithms. In 2020, Beullens [23] proposed two new attacks on Rainbow; the intersection attack which also works on the UOV scheme and the rectangular MinRank attack. Both attacks greatly reduce the key recovery cost and in consequence, the parameter sets fail to meet the security requirements set out by NIST. Additionally, another Beullens' key recovery attack that completely breaks the Rainbow scheme on security SL1 disqualified it from making it to Round 4 [24]. Despite the total break of the Rainbow scheme, Cartor et al. [25] suggested adding an internal perturbation modifier in order to mend the scheme and make it secure again.

2. Related Works

The concept of an equivalent public key was first introduced in [26] where they generalized equivalent keys to increase the efficiency of algebraic key recovery attacks. [27] implemented the concept of an equivalent public key to give a detailed security analysis of their proposed encryption scheme. Furthermore, [28] showed that the algebraic system of an EFC public key has lower degree equations during the Gröbner basis computation compared to a random system having the same size. Consequently, solving the algebraic system of an EFC public key becomes simpler and easier.

We aim to construct the weakened multivariate signature schemes by focusing on generating the public–private key pair of which its vulnerability is only known to the RCA. The public key system $\mathcal{P}$ of multivariate signature schemes will be constructed by inducing some weaknesses but still inherits randomness. Furthermore, we put forward strategies to identify them so that the users could conduct due diligence upon receiving the key pair.

In this work, we provide three potential methodologies that could be executed by an RCA which will expose UOV and Rainbow signature schemes to forgery. All three methodologies are able to forge the UOV signature scheme, whilst the Rainbow signature scheme is only vulnerable to one methodology. We also discuss the reason why the Rainbow signature scheme is secure against the first and second forgery mechanisms. In addition, we provide two strategies to identify whether the public key of UOV and Rainbow signature schemes obtained from a potential RCA, has the potential to be utilized to forge signatures. Consequently, the users of the UOV and Rainbow signature schemes can refuse to use the key pairs from the RCA.

The layout of the paper is structured as follows. In Section 3, we summarize the UOV and Rainbow signature schemes. Section 4 summarizes the three forgery mechanisms denoted by DSFM1, DSFM2 and DSFM3. We also discuss methods to identify whether one is provided weak parameters via DSFM1 and DSFM2 methodologies. Moreover, in Section 5, we present our main results which show that the UOV scheme is vulnerable against the mentioned forgery mechanisms. We also provide examples for illustrative purposes. Moving on to Section 6, we discuss the reason why the Rainbow scheme is not vulnerable to DSFM1 and DSFM2. Next, we show that both UOV and Rainbow schemes are vulnerable against DSFM3 in Section 7. Section 8 provides the discussion from our work for comprehensive understanding. Finally, we conclude our work in Section 9.

3. Multivariate Signature Schemes

In this section, we show the key generation, signing and verification processes of two multivariate signature schemes, namely the UOV and Rainbow signature schemes.

The UOV signature scheme can be described as follow.

3.1. UOV Digital Signature

Let $\mathbb{F}_q$ be a finite field with q elements. The number of equations is equal to o and the number of variables is equal to $n = o + v$ where $v > o$. Let $V = 1, \dots, v$ and $O = v + 1, \dots, n$. $x_1, \dots, x_v$ be known as the Vinegar variables and $x_{v+1}, \dots, x_n$ known as the Oil variables.

Key Generation: Choose an affine map $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$ and a central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^o$ which consists of o quadratic polynomials $f^{(1)}, \dots, f^{(o)}$ of the form

$$f^{(k)} = \sum_{a,b \in V} \alpha_{a,b}^{(k)} x_a x_b + \sum_{a \in V, b \in O} \beta_{a,b}^{(k)} x_a x_b + \sum_{a \in V \cup O} \gamma_a^{(k)} x_a + \delta^{(k)} (k = 1, \dots, o).$$

The *private key* consists of the two maps $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^o$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$, whereas the *public key* is the composed map $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ consisting of o quadratic polynomials in n variables.

Signature Generation: To generate a signature $\mathbf{z} \in \mathbb{F}^n$ for a document d, one uses a hash function $\mathcal{H} : \{0, 1\} \to \mathbb{F}^o$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^o$ and perform the following steps.

1. Find a pre-image $\mathbf{y} \in \mathbb{F}^n$ of $\mathbf{w}$ under the central map $\mathcal{F}$.
   - Choose random values for the Vinegar variables $y_1, \dots, y_v$ and substitute them into the polynomials $f^{(1)}, \dots, f^{(o)}$.
   - Choose the resulting linear system of o equations in the o Oil variables $y_{v+1}, \dots, y_n$ by Gaussian elimination. If the system does not have a solution, choose other values for the vinegar variables $x_1, \dots, x_v$ and try again.
2. Compute the signature $\mathbf{z} \in \mathbb{F}^n$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Signature Verification: To check if $\mathbf{z} \in \mathbb{F}^n$ is indeed a valid signature for the document d, one computes $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^o$ and computes $\mathbf{w}' = \mathcal{P}(\mathbf{z}) = \mathbb{F}^o$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature $\mathbf{z}$ is accepted, otherwise rejected.

Next, we describe the Rainbow signature scheme as follows.

3.2. Rainbow Digital Signature

Key Generation: Let $\mathbb{F}_q$ be a finite field with q elements. Let $v_1, \dots, v_{u+1}$ be integers such that $0 < v_1 < v_2, \dots < v_u < v_{u+1} = n$ and define the sets of integers $V_i = \{1, \dots, v_i\}$ for $i = 1, \dots, u$. We set $o_i = v_{i+1} - v_i$ and $O_i = \{v_i + 1, \dots, v_{i+1}\}$ for $i = 1, \dots, u$. We have $|O_i| = o_i$.

The central map $\mathcal{F}$ consists of $m = n - v_1$ polynomials $f^{(v_1+1)}, \dots, f^{(n)} \in \mathbb{F}[x_1, \dots, x_n]$ of the form

$$f^{(k)}(\mathbf{x}) = \sum_{a,b \in V_\ell, a \leq b} \alpha_{a,b}^{(k)} x_a x_b + \sum_{a \in O_\ell, b \in V_\ell} \beta_{a,b}^{(k)} x_a x_b + \sum_{a \in V_\ell \cup O_\ell} \gamma_a^{(k)} x_a + \eta^{(k)} (k = v_1 + 1, \dots, n),$$

where ℓ is the only integer such that $k \in O_\ell$.

To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. Hence, the *public key* has the form $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$, the *private key* consists of the three maps $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$.

The following Algorithm 1 is to compute the inversion of the Rainbow central map.

Algorithm 1 Inversion of the Rainbow central map.

```
Input: Rainbow central map $\mathcal{F} = (f^{(v_1+1)}, \ldots, f^{(n)}$, vector $\mathbf{x} \in \mathbb{F}^m$.
Output: Vector $\mathbf{y} \in \mathbb{F}^n$ with $\mathcal{F}(\mathbf{y}) = \mathbf{x}$.
1. Choose random values for the variables $y_1, \ldots, y_{v_1}$ and substitute them into the polynomials $f^{(i)}(i = v_1 + 1, \ldots, n)$.
2. for $\ell = 1$ to $u$ do
3.     Perform Gaussian Elimination on the polynomials $f^{(i)}(i \in O_\ell)$ to get the values of the variables $x_i(i \in O_\ell)$.
4.     Substitute the values of $x_i(i \in O_\ell)$ into the polynomials $f^{(i)}(i = v_{\ell+1} + 1, \ldots, n)$.
5. end for
```

Signature Generation: To generate a signature for a message d, one uses a hash function $\mathcal{H} : \{0, 1\} \to \mathbb{F}^m$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ and perform the following steps.

1. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^m$.
2. Compute a pre-image $\mathbf{y} \in \mathbb{F}^n$ of $\mathbf{x}$ under the central map $\mathcal{F}$. This is done utilizing Algorithm 1.
3. Compute the signature $\mathbf{z} \in \mathbb{F}^n$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Signature Verification: To check, if $\mathbf{z} \in \mathbb{F}^n$ is indeed a valid signature for the document d, one computes $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ and computes $\mathbf{w}' = \mathcal{P}(\mathbf{z}) = \mathbb{F}^m$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature $\mathbf{z}$ is accepted, otherwise rejected.

4. Novel Forgery Mechanisms for Multivariate Signature Schemes

This section outlines two forgery mechanisms which were first made known to the public by our research group during The International Conference on Mathematical Sciences and Technology 2022 (MathTech 2022) [29] namely the DSFM1 and DSFM2 mechanisms. We also provide another novel mechanism in this section, known as DSFM3. Upon executing these three methods on multivariate signature schemes, it would enable an adversary to forge signatures without the knowledge of $(\mathcal{S}, \mathcal{F}, \mathcal{T})$.

4.1. Digital Signature Forgery Mechanism 1 (DSFM1)

In this subsection, a public key system $\mathcal{P}$ which is generated by DSFM1 would enable forgery by those who know about it. The DSFM1 would result in polynomials in $\mathcal{P}$ to be multiples of each other. As such, one needs to solve only one of the polynomials $p^{(i)}(i = 1, \ldots, m)$. This is due to the fact that a vector $\mathbf{x} = (x_1, \ldots, x_m)$ which satisfies $p^{(i)}(\mathbf{x}) = 0$ also satisfies the other polynomials in the same system $\mathcal{P}$.

4.1.1. Generating DSFM1 Induced System of Equations

The following Algorithm 2 induces DSFM1 weaknesses on a system of equations.

Algorithm 2 Digital Signature Forgery Mechanism 1

```
Input: Integer $q$.
Output: Public key system $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ over $\mathbb{F}_q$.
1. Choose two random invertible affine maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$.
2. Choose a central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ of which its polynomials can also be written as
   $f^{(j)} = k_j f^{(1)}$ where $k_j \in \mathbb{Z}_q$.
3. Compute $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$.
```

4.1.2. Identifying DSFM1

The user can check whether the public key system $\mathcal{P}$ received is a forgeable system via DSFM1 or not by utilizing the following Algorithm 3.

Algorithm 3 Identifying DSFM1

```
Input: The system $\mathcal{P} = (p^{(1)}, \ldots, p^{(m)})$ of multivariate quadratic polynomials over $\mathbb{F}_q$
Output: $\mathcal{P}$ is a forgeable system
1. for $j = 2$ to $m$ do
2.     $k_j = c_{p^{(j)}} \cdot c^{-1}_{p^{(1)}} \text{mod } q$ where $c_{p^{(j)}}$ and $c_{p^{(1)}}$ are the coefficients of polynomial $p^{(j)}$
   and $p^{(1)}$, respectively.
3.     If $p^{(j)} = k_j p^{(1)}$ where $k_j \in \mathbb{Z}_q$, then $\mathcal{P}$ is a forgeable system.
4. end for
5. return
```

4.2. Digital Signature Forgery Mechanism 2 (DSFM2)

In this subsection, a public key system $\mathcal{P}$ which is generated by DSFM2 would enable forgery by those who know about it. The DSFM2 would result in polynomials in $\mathcal{P}$ to be summations of each other. As such, one needs to solve only two of the polynomials $p^{(i)}$ and $p^{(k)}$ where $p^{(j)} = p^{(i)} + p^{(k)}$. This is due to the fact that a vector $\mathbf{x} = (x_1, \ldots, x_m)$ which satisfies $p^{(i)}(\mathbf{x}) = 0$ and $p^{(k)}(\mathbf{x}) = 0$ also satisfies the other polynomials in the same system $\mathcal{P}$.

4.2.1. Generating DSFM2 Induced System of Equations

The following Algorithm 4 induces DSFM2 weaknesses in a system of equations.

Algorithm 4 Digital Signature Forgery Mechanism 2

```
Input: Integer $q$.
Output: Public key system $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ over $\mathbb{F}_q$.
1. Choose two secret invertible affine maps $\mathcal{S} : \mathbb{F}^2 \to \mathbb{F}^2$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$.
2. Choose a secret central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^2$.
3. Compute $\mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$ to output two equations $p^{(1)}$ and $p^{(2)}$. For $p^{(j)}$ $(j = 3, \ldots, m)$, set
   $p^{(j)} = p^{(i)} + p^{(k)}$ where $i = 1, \ldots, j-1$ and $k = 1, \ldots, j-1$.
4. Publish $\mathcal{P} = (p^{(1)}, \ldots, p^{(m)})$ as public key over $\mathbb{F}_q$.
```

4.2.2. Identifying DSFM2

The user can check whether the public key system $\mathcal{P}$ received is a forgeable system via DSFM2 or not by utilizing the following Algorithm 5.

Algorithm 5 Identifying DSFM2

```
Input: The system $\mathcal{P} = (p^{(1)}, \ldots, p^{(m)})$ of multivariate quadratic polynomials over $\mathbb{F}_q$
Output: $\mathcal{P}$ is a forgeable system
for $j = 3$ to $m$ do
    for $i = 1$ to $j - 1$ do
        for $k = i$ to $j - 1$ do
            $p^{(i)} + p^{(k)}$
            If $p^{(j)} = p^{(i)} + p^{(k)}$, then $\mathcal{P}$ is a forgeable system.
        end for
    end for
end for
return
```

4.3. Digital Signature Forgery Mechanism 3 (DSFM3)

In this subsection, we discuss the method to forge multivariate signature schemes without having to alter the construction of the public key system $\mathcal{P}$. This is due to the fact that if an adversary successfully solicits $\mathbf{x}$ from an RCA and solves $\mathcal{P}(\mathbf{x} + \alpha) = \mathbf{w}$ for some $\alpha \in \mathbb{Z}_q$, the adversary can forge the signature $\mathbf{z}$ corresponding to the hash value $\mathbf{w} = \mathcal{H}(d)$.

Generating DSFM3 Forged Signature

The following Algorithm 6 explains DSFM3.

Algorithm 6 Digital Signature Forgery Mechanism 3

```
Input: Public key $\mathcal{P} = (p^{(1)}, \ldots, p^{(m)})$, $\mathbf{x}$= $(x_1, \ldots, x_n)$ such that $\mathcal{P}(\mathbf{x}) = 0$ and $\mathbf{w}$= $(w_1, \ldots, w_m)$
Output: Signature $\mathbf{z}'$ such that $\mathcal{P}(\mathbf{z}') = \mathbf{w}' = \mathbf{w}$
1. Compute $\mathcal{P}(\mathbf{x} + \alpha) = (p^{(1)}(x_1 + \alpha, \ldots, x_n + \alpha), \ldots, p^{(m)}(x_1 + \alpha, \ldots, x_n + \alpha))$ where $\alpha$
   is an unknown variable.
2. Solve $m$ equations in the single variable $\alpha$ such that $\mathcal{P}(\mathbf{x} + \alpha) = \mathbf{w}$.
3. Set $\mathbf{z}' = \mathbf{x}' + \alpha = (x_1 + \alpha, \ldots, x_n + \alpha)$.
```

In Steps 1 and 2, computing and solving $\mathcal{P}(\mathbf{x} + \alpha) = \mathbf{w}$ would reduce the number of unknowns from n variables to only one variable. Instead of solving m equations in n variables, the adversary only needs to solve m univariable equations which is much easier.

5. Generating Weak UOV Signature Scheme

In this section, we show how a weak UOV signature scheme is generated by RCA from DSFM1 and DSFM2.

5.1. Generating Weak UOV Signature Scheme by DSFM1

From DSFM1, we put forward an algorithm to generate a weak UOV public key. In other words, we set up the UOV public key which is $\mathcal{P}$, where all of its polynomials satisfy the original form and also can be written into multiples of each other. The following Algorithm 7 explains the key generation of weak UOV signature scheme by DSFM1.

```
Algorithm 7 Key Generation of Weak UOV Signature Scheme by DSFM1
Input: Integers $o$ and $v$ such that $v > o$ and $n = o + v$. Let $V = 1, \ldots, v$ and $O = v + 1, \ldots, n$. Let $x_1, \ldots, x_v$ be the Vinegar variables and $x_{v+1}, \ldots, x_n$ be the Oil variables.
Output: Public key $\mathcal{P}$ in the form of $p^{(j)} = k_j p^{(1)}$ for $j = 2, \ldots, o$.
1. Choose a random invertible affine map $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$.
2. Choose a random polynomial $f^{(1)}$ of the form
       $f^{(1)} = \sum_{a,b \in V} \alpha^{(1)}_{a,b} x_a x_b + \sum_{a \in V, b \in O} \beta^{(1)}_{a,b} x_a x_b + \sum_{a \in V \cup O} \gamma^{(1)}_a x_a + \delta^{(1)}$
   and for $j = 2, \ldots, m$ compute $f^{(j)} = k_j f^{(1)}$ where $k_j \in \mathbb{Z}_q$. Set the central map $\mathcal{F} = (f^{(1)}, \ldots, f^{(o)}) : \mathbb{F}^n \to \mathbb{F}^o$.
3. Compute $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$. The system $\mathcal{P} = (p^{(1)}, \ldots, p^{(o)})$ consists of $o$ quadratic polynomials in $n$ variables.
```

To pass through the verification, the vectors in $\mathbf{w}$ must be multiple to each other, otherwise the verification fails. This is because, the polynomials in public key system $\mathcal{P}$ and the central map $\mathcal{F}$ are of the form $p^{(j)} = k_j p^{(1)}$ and $f^{(j)} = k_j f^{(1)}$, respectively. The following Algorithm 8 explains the signature generation of weak UOV signature scheme by DSFM1.

```
Algorithm 8 Signature Generation of Weak UOV Signature Scheme by DSFM1
Input: Document $d$
Output: Signature $\mathbf{z}$
1. Compute $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^o$ such that $w_j = k_j w_1$ $(j = 2, \ldots, o)$.
2. Find a pre-image $\mathbf{y} \in \mathbb{F}^n$ of $\mathbf{w}$ under the central map $\mathcal{F}$.
   • Choose random values for the Vinegar variables $y_1, \ldots, y_v$ and substitute them into the polynomials $f^{(1)}, \ldots, f^{(o)}$.
   • Choose the resulting linear system of $o$ equations in the $o$ Oil variables $y_{v+1}, \ldots, y_n$ by Gaussian elimination. If the system does not have a solution, choose other values for the vinegar variables $x_1, \ldots, x_v$ and try again.
3. Compute the signature $\mathbf{z} \in \mathbb{F}^n$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.
```

The signature verification of the UOV signature scheme generated by DSFM1 as in Algorithm 9 below works the same as the original UOV.

```
Algorithm 9 Signature Verification of Weak UOV Signature Scheme by DSFM1
Input: Public key $\mathcal{P}$, document $d$ and signature $\mathbf{z}$
Output: Accept or reject signature
1. Compute $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^o$
2. Compute $\mathbf{w}' = \mathcal{P}(\mathbf{z})$
3. If $\mathbf{w}' = \mathbf{w}$ holds, the signature $\mathbf{z}$ is accepted, otherwise rejected.
```

In the following example, we illustrate the generation of a weak UOV scheme via the DSFM1 methodology as well as the signing and verification process. The example below shows that a weak UOV scheme can still be used by a user without suspicion since the constants seem randomized and the signing and verification work as normal.

Example 1. *We will discuss key generation, signing and verification on* $\mathbb{F} = GF(7)$*.*

Key Generation: We choose $(o, v) = (3, 5)$, which will lead to a public key of 3 quadratic equations in 8 variables. The private key consists of the affine map $\mathcal{T} : \mathbb{F}^8 \to \mathbb{F}^8$.

$$\mathcal{T}(x_1, \ldots, x_8) = \begin{pmatrix} 2 & 1 & 5 & 3 & 1 & 0 & 3 & 2 \\ 4 & 4 & 1 & 6 & 2 & 1 & 4 & 2 \\ 3 & 5 & 3 & 2 & 1 & 6 & 4 & 0 \\ 5 & 5 & 3 & 5 & 6 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 4 & 2 & 5 & 5 \\ 3 & 1 & 1 & 5 & 1 & 0 & 6 & 2 \\ 1 & 1 & 2 & 1 & 6 & 5 & 2 & 3 \\ 0 & 3 & 4 & 1 & 6 & 5 & 6 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 4 \\ 3 \\ 6 \\ 5 \\ 4 \\ 2 \end{pmatrix}$$

and the central map $\mathcal{F} : \mathbb{F}^8 \to \mathbb{F}^3$ is given by polynomials

$$\begin{aligned}
f^{(1)} = {} & x_1^2 + x_1x_2 + 3x_1x_3 + 2x_1x_4 + 6x_1x_5 + 4x_1x_6 + 2x_1x_7 + 5x_1x_8 + 3x_2^2 + x_2x_3 + 5x_2x_4 \\
& + x_2x_6 + 3x_2x_7 + 3x_2x_8 + 4x_3x_4 + 6x_3x_6 + 2x_3x_7 + 5x_3x_8 + 4x_4x_5 + 5x_4x_6 + 3x_4x_7 \\
& + 2x_4x_8 + 3x_2 + 6x_3 + 3x_4 + 5x_5 + 6x_6 + 2x_7 + 2x_8 + 4 \\
f^{(2)} = {} & 2f^{(1)} \pmod 7 \\
= {} & 2x_1^2 + 2x_1x_2 + 6x_1x_3 + 4x_1x_4 + 5x_1x_5 + x_1x_6 + 4x_1x_7 + 3x_1x_8 + 6x_2^2 + 2x_2x_3 + 3x_2x_4 \\
& + 2x_2x_6 + 6x_2x_7 + 6x_2x_8 + x_3x_4 + 5x_3x_6 + 4x_3x_7 + 3x_3x_8 + x_4x_5 + 3x_4x_6 + 6x_4x_7 \\
& + 4x_4x_8 + 6x_2 + 5x_3 + 6x_4 + 3x_5 + 5x_6 + 4x_7 + 4x_8 + 1 \\
f^{(3)} = {} & 5f^{(1)} \pmod 7 \\
= {} & 5x_1^2 + 5x_1x_2 + x_1x_3 + 3x_1x_4 + 2x_1x_5 + 6x_1x_6 + 3x_1x_7 + 4x_1x_8 + x_2^2 + 5x_2x_3 + 4x_2x_4 \\
& + 5x_2x_6 + x_2x_7 + x_2x_8 + 6x_3x_4 + 2x_3x_6 + 3x_3x_7 + 4x_3x_8 + 6x_4x_5 + 4x_4x_6 + x_4x_7 \\
& + 3x_4x_8 + x_2 + 2x_3 + x_4 + 4x_5 + 2x_6 + 3x_7 + 3x_8 + 6
\end{aligned}$$

We compute the public key $\mathcal{P} = (p^{(1)}, p^{(2)}, p^{(3)}) : \mathbb{F}^8 \to \mathbb{F}^3$ by $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$, which results in

$$\begin{aligned}
p^{(1)} = {} & 3x_1x_2 + 2x_1x_3 + 5x_1x_4 + 6x_1x_5 + x_1x_6 + 5x_1x_7 + 6x_1x_8 + 3x_2^2 + 6x_2x_3 + 2x_2x_4 + 6x_2x_5 \\
& + 6x_3^2 + 5x_3x_4 + x_3x_5 + 6x_3x_6 + 6x_3x_8 + 6x_4^2 + 4x_4x_6 + 6x_4x_8 + 5x_5^2 + 2x_5x_6 + 4x_5x_7 \\
& + 2x_5x_8 + 2x_6^2 + x_6x_7 + 5x_6x_8 + 2x_7^2 + 3x_7x_8 + 2x_8^2 + 3x_1 + 4x_2 + 2x_3 + x_4 + 4x_5 + 3x_6 \\
& + 4x_7 + 2x_8 \\
p^{(2)} = {} & 6x_1x_2 + 4x_1x_3 + 3x_1x_4 + 5x_1x_5 + 2x_1x_6 + 3x_1x_7 + 5x_1x_8 + 6x_2^2 + 5x_2x_3 + 4x_2x_4 + 5x_2x_5 \\
& + 5x_3^2 + 3x_3x_4 + 2x_3x_5 + 5x_3x_6 + 5x_3x_8 + 5x_4^2 + x_4x_6 + 5x_4x_8 + 3x_5^2 + 4x_5x_6 + x_5x_7 \\
& + 4x_5x_8 + 4x_6^2 + 2x_6x_7 + 3x_6x_8 + 4x_7^2 + 6x_7x_8 + 4x_8^2 + 6x_1 + x_2 + 4x_3 + 2x_4 + x_5 + 6x_6 \\
& + x_7 + 4x_8 \\
p^{(3)} = {} & x_1x_2 + 3x_1x_3 + 4x_1x_4 + 2x_1x_5 + 5x_1x_6 + 4x_1x_7 + 2x_1x_8 + x_2^2 + 2x_2x_3 + 3x_2x_4 + 2x_2x_5 \\
& + 2x_3^2 + 4x_3x_4 + 5x_3x_5 + 2x_3x_6 + 2x_3x_8 + 2x_4^2 + 6x_4x_6 + 2x_4x_8 + 4x_5^2 + 3x_5x_6 + 6x_5x_7 \\
& + 3x_5x_8 + 3x_6^2 + 5x_6x_7 + 4x_6x_8 + 3x_7^2 + x_7x_8 + 3x_8^2 + x_1 + 6x_2 + 3x_3 + 5x_4 + 6x_5 + x_6 \\
& + 6x_7 + 3x_8.
\end{aligned}$$

Signature Generation: In order to generate a signature for the message $\mathbf{w} = (3, 6, 1)$, we first need to compute $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{w})$. We choose random values for the Vinegar variables $(x_1, x_2, x_3, x_4, x_5) = (3, 3, 6, 1, 2)$ and substitute them into the polynomials $f^{(1)}, f^{(2)}$ and $f^{(3)}$. Thus, we obtain a linear system in the Oil variables x_6, x_7 and x_8 of the form

$$\begin{aligned}
\bar{f}^{(1)} &= 6x_6 + 4x_7 + 2x_8 + 6\\
\bar{f}^{(2)} &= 5x_6 + x_7 + 4x_8 + 4\\
\bar{f}^{(3)} &= 2x_6 + 6x_7 + 3x_8 + 3.
\end{aligned}$$

By Gaussian elimination, this system has the solution $(x_6, x_7, x_8) = (0, 4, 3)$. Attaching the Vinegar variables yields

$$\mathbf{y} = \mathcal{F}^{-1}(\mathbf{w}) = (3, 3, 6, 1, 2, 0, 4, 3).$$

Finally, we compute

$$\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y}) = (5, 5, 3, 2, 5, 4, 1, 0)$$

to obtain a signature $\mathbf{z} \in \mathbb{F}^8$ for the message $\mathbf{w}$.

Signature Verification: In order to check if $\mathbf{z}$ is indeed a valid signature for the message $\mathbf{w}$, we compute

$$\mathbf{w}' = \mathcal{P}(\mathbf{z}) = (3, 6, 1).$$

Since $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted.

5.1.1. A Weakened DSFM1 UOV Signature Scheme Forgery Methodology

The algorithm to forge the signature of a weak UOV scheme by DSFM1 is described in Algorithm 10 as below.

Algorithm 10 Forgery of Weakened DSFM1 UOV Signature Scheme

```
Input: Public key $\mathcal{P}$, document $d$
Output: Signature $\mathbf{z}'$ such that $\mathcal{P}(\mathbf{z})' = \mathbf{w}' = \mathbf{w}$
1.  Solve $p^{(1)}(x) = 0$ and obtain $\mathbf{z}' = (z_1, \ldots, z_o)$.
```

Since $p^{(j)} = k_j p^{(1)}$, solving one of the polynomials would solve the whole system $\mathcal{P}$.

In the following example, we show how an impersonator successfully forge the signature of a weakened DSFM1 UOV scheme.

Example 2. *Given the public key $\mathcal{P} = (p^{(1)}, p^{(2)}, p^{(3)})$ of a weakened DSFM1 UOV scheme as in Example 1:*

$$\begin{aligned}
p^{(1)} = {} & 3x_1x_2 + 2x_1x_3 + 5x_1x_4 + 6x_1x_5 + x_1x_6 + 5x_1x_7 + 6x_1x_8 + 3x_2^2 + 6x_2x_3 + 2x_2x_4 + 6x_2x_5\\
& + 6x_3^2 + 5x_3x_4 + x_3x_5 + 6x_3x_6 + 6x_3x_8 + 6x_4^2 + 4x_4x_6 + 6x_4x_8 + 5x_5^2 + 2x_5x_6 + 4x_5x_7\\
& + 2x_5x_8 + 2x_6^2 + x_6x_7 + 5x_6x_8 + 2x_7^2 + 3x_7x_8 + 2x_8^2 + 3x_1 + 4x_2 + 2x_3 + x_4 + 4x_5 + 3x_6\\
& + 4x_7 + 2x_8\\
p^{(2)} = {} & 6x_1x_2 + 4x_1x_3 + 3x_1x_4 + 5x_1x_5 + 2x_1x_6 + 3x_1x_7 + 5x_1x_8 + 6x_2^2 + 5x_2x_3 + 4x_2x_4 + 5x_2x_5\\
& + 5x_3^2 + 3x_3x_4 + 2x_3x_5 + 5x_3x_6 + 5x_3x_8 + 5x_4^2 + x_4x_6 + 5x_4x_8 + 3x_5^2 + 4x_5x_6 + x_5x_7\\
& + 4x_5x_8 + 4x_6^2 + 2x_6x_7 + 3x_6x_8 + 4x_7^2 + 6x_7x_8 + 4x_8^2 + 6x_1 + x_2 + 4x_3 + 2x_4 + x_5 + 6x_6\\
& + x_7 + 4x_8\\
p^{(3)} = {} & x_1x_2 + 3x_1x_3 + 4x_1x_4 + 2x_1x_5 + 5x_1x_6 + 4x_1x_7 + 2x_1x_8 + x_2^2 + 2x_2x_3 + 3x_2x_4 + 2x_2x_5\\
& + 2x_3^2 + 4x_3x_4 + 5x_3x_5 + 2x_3x_6 + 2x_3x_8 + 2x_4^2 + 6x_4x_6 + 2x_4x_8 + 4x_5^2 + 3x_5x_6 + 6x_5x_7\\
& + 3x_5x_8 + 3x_6^2 + 5x_6x_7 + 4x_6x_8 + 3x_7^2 + x_7x_8 + 3x_8^2 + x_1 + 6x_2 + 3x_3 + 5x_4 + 6x_5 + x_6\\
& + 6x_7 + 3x_8.
\end{aligned}$$

Let $\mathbf{w}=(3,6,1)$*. The impersonator computes* $\bar{p}^{(i)}=p^{(i)}-w_i \pmod 7$ *where* $i=1,2,3$*. Then, to solve* $\bar{p}^{(1)}(x)=0$*, he chooses random variables* $(x_1,x_2,x_3,x_4,x_5,x_6,x_7)=(4,5,2,5,4,3,3)$ *and substitutes them into* $\bar{p}^{(1)}$ *which results a quadratic equation with one variable*

$$2x_8^2+2x_8+3=0.$$

Since this equation has two solutions $x_8=1$ *and* 5 *hence the solution for* $p^{(1)}$ *are* $(4,5,2,5,4,3,3,1)$ *and* $(4,5,2,5,4,3,3,5)$*. These solutions are also the solutions to* $p^{(2)}$ *and* $p^{(3)}$*, which implies* $\mathcal{P}(\mathbf{z}')=\mathbf{w}$ *where* $\mathbf{z}'=(4,5,2,5,4,3,3,1)$ *or* $\mathbf{z}'=(4,5,2,5,4,3,3,5)$ *are the forged signatures. Indeed both* $\mathbf{z}'\neq\mathbf{z}=(5,5,3,2,5,4,1,0)$*.*

5.1.2. Identifying a Weakened DSFM1 UOV Scheme

We can directly use Algorithm 3 to identify a weakened DSFM1 UOV scheme.

Example 3. *Given the public key* $\mathcal{P}=(p^{(1)},p^{(2)},p^{(3)})$ *of a weakened DSFM1 UOV scheme as in Example 1:*

$$\begin{aligned}
p^{(1)} = {} & 3x_1x_2+2x_1x_3+5x_1x_4+6x_1x_5+x_1x_6+5x_1x_7+6x_1x_8+3x_2^2+6x_2x_3+2x_2x_4+6x_2x_5\\
&+6x_3^2+5x_3x_4+x_3x_5+6x_3x_6+6x_3x_8+6x_4^2+4x_4x_6+6x_4x_8+5x_5^2+2x_5x_6+4x_5x_7\\
&+2x_5x_8+2x_6^2+x_6x_7+5x_6x_8+2x_7^2+3x_7x_8+2x_8^2+3x_1+4x_2+2x_3+x_4+4x_5+3x_6\\
&+4x_7+2x_8\\
p^{(2)} = {} & 6x_1x_2+4x_1x_3+3x_1x_4+5x_1x_5+2x_1x_6+3x_1x_7+5x_1x_8+6x_2^2+5x_2x_3+4x_2x_4+5x_2x_5\\
&+5x_3^2+3x_3x_4+2x_3x_5+5x_3x_6+5x_3x_8+5x_4^2+x_4x_6+5x_4x_8+3x_5^2+4x_5x_6+x_5x_7\\
&+4x_5x_8+4x_6^2+2x_6x_7+3x_6x_8+4x_7^2+6x_7x_8+4x_8^2+6x_1+x_2+4x_3+2x_4+x_5+6x_6\\
&+x_7+4x_8\\
p^{(3)} = {} & x_1x_2+3x_1x_3+4x_1x_4+2x_1x_5+5x_1x_6+4x_1x_7+2x_1x_8+x_2^2+2x_2x_3+3x_2x_4+2x_2x_5\\
&+2x_3^2+4x_3x_4+5x_3x_5+2x_3x_6+2x_3x_8+2x_4^2+6x_4x_6+2x_4x_8+4x_5^2+3x_5x_6+6x_5x_7\\
&+3x_5x_8+3x_6^2+5x_6x_7+4x_6x_8+3x_7^2+x_7x_8+3x_8^2+x_1+6x_2+3x_3+5x_4+6x_5+x_6\\
&+6x_7+3x_8.
\end{aligned}$$

To identify $\mathcal{P}$ *is a forgeable system, we choose one coefficient from* $p^{(1)},p^{(2)}$ *and* $p^{(3)}$ *and compute*

$$\begin{aligned}
k_2 &= 6\times 3^{-1} \pmod 7 = 2\\
k_3 &= 1\times 3^{-1} \pmod 7 = 5.
\end{aligned}$$

Since

$$\begin{aligned}
p^{(2)} &= 2\times p^{(1)} \pmod 7\\
p^{(3)} &= 5\times p^{(1)} \pmod 7
\end{aligned}$$

is true, we have successfully identified that $\mathcal{P}$ *is a forgeable system.*

5.2. Generating Weak UOV Signature Scheme by DSFM2

From DSFM2, we put forward an algorithm to generate a weak UOV public key. In other words, we set up the UOV public key, which is $\mathcal{P}$, where all its polynomials satisfy the original form and also can be written into summation of two polynomials from the same system. The following Algorithm 11 explains the key generation of weak UOV signature scheme by DSFM2.

```
**Algorithm 11** Key Generation of Weak UOV Signature Scheme by DSFM2
**Input:** Integers $o$ and $v$ such that $v > o$ and $n = o + v$. Let $V = 1, \ldots, v$ and $O = v + 1, \ldots, n$.
Let $x_1, \ldots, x_v$ be the Vinegar variables and $x_{v+1}, \ldots, x_n$ be the Oil variables.
**Output:** Public key $\mathcal{P}$ in the form of $p^{(j)} = p^{(i)} + p^{(k)}$ for $j = 3, \ldots, o$, $i = 1, \ldots, j-1$ and
$k = 1, \ldots, j-1$.
1. Choose a random invertible affine map $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$.
2. Choose a random polynomial $f^{(1)}(x) = 0$ and $f^{(2)}(x) = 0$ of the form
       $$f^{(1,2)} = \sum_{a,b \in V} \alpha_{a,b}^{(1,2)} x_a x_b + \sum_{a \in V, b \in O} \beta_{a,b}^{(1,2)} x_a x_b + \sum_{a \in V \cup O} \gamma_a^{(1,2)} x_a + \delta^{(1,2)}.$$
   For $j = 3, \ldots, o$ compute $f^{(j)} = f^{(i)} + f^{(k)}$ where $i = 1, \ldots, j-1$ and $k = 1, \ldots, j-1$.
   Set the central map $\mathcal{F} = (f^{(1)}, \ldots, f^{(o)}) : \mathbb{F}^n \to \mathbb{F}^o$.
3. Compute $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$. The system $\mathcal{P}$ consists of $o$ quadratic polynomials in $n$ variables.
```

To pass through the verification, the vectors in $\mathbf{w}$ must be of the form $w_j = w_i + w_k$, otherwise the verification fails. This is because, the polynomials in public key system $\mathcal{P}$ and the central map $\mathcal{F}$ are of the form $p^{(j)} = p^{(i)} + p^{(k)}$ and $f^{(j)} = f^{(i)} + f^{(k)}$, respectively. The following Algorithm 12 explains the signature generation of weak UOV signature scheme by DSFM2.

```
**Algorithm 12** Signature Generation of Weak UOV Signature Scheme by DSFM2
**Input:** Document $d$
**Output:** Signature $\mathbf{z}$
1. Compute $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^o$ such that $w_j = w_i + w_k$ for $j = 3, \ldots, o$, $i = 1, \ldots, j-1$,
   $k = 1, \ldots, j-1$.
2. Find a pre-image $\mathbf{y} \in \mathbb{F}^n$ of $\mathbf{w}$ under the central map $\mathcal{F}$.
   • Choose random values for the Vinegar variables $y_1, \ldots, y_v$ and substitute them
     into the polynomials $f^{(1)}, \ldots, f^{(o)}$.
   • Choose the resulting linear system of $o$ equations in the $o$ Oil variables
     $y_{v+1}, \ldots, y_n$ by Gaussian elimination. If the system does not have a solution,
     choose other values for the vinegar variables $x_1, \ldots, x_v$ and try again.
3. Compute the signature $\mathbf{z} \in \mathbb{F}^n$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.
```

The signature verification of the UOV signature scheme generated by DSFM1 as in Algorithm 13 below works the same as the original UOV.

```
**Algorithm 13** Signature Verification of Weak UOV Signature Scheme by DSFM2
**Input:** Public key $\mathcal{P}$, document $d$ and signature $\mathbf{z}$
**Output:** Accept or reject signature
1. Compute $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^o$.
2. Compute $\mathbf{w}' = \mathcal{P}(\mathbf{x})$.
3. If $\mathbf{w}' = \mathbf{w}$ holds, the signature $\mathbf{z}$ is accepted, otherwise rejected.
```

In the following example, we illustrate the generation of a weak UOV scheme via the DSFM2 methodology as well as the signing and verification process. The example below shows that a weak UOV scheme can still be used by a user without suspicion since the constants seem randomized and the signing and verification work as normal.

Example 4. *We will discuss key generation, signing and verification on $\mathbb{F} = GF(7)$.*

Key Generation: We choose $\mathbb{F} = GF(7)$, and $(o, v) = (3, 5)$, which will lead to a public key of 3 quadratic equations in 8 variables. The private key consists of the affine map $\mathcal{T} : \mathbb{F}^8 \to \mathbb{F}^8$.

$$\mathcal{T}(x_1, \ldots, x_8) = \begin{pmatrix} 2 & 1 & 5 & 3 & 1 & 0 & 3 & 2 \\ 4 & 4 & 1 & 6 & 2 & 1 & 4 & 2 \\ 3 & 5 & 3 & 2 & 1 & 6 & 4 & 0 \\ 5 & 5 & 3 & 5 & 6 & 2 & 3 & 4 \\ 1 & 0 & 1 & 2 & 4 & 2 & 5 & 5 \\ 3 & 1 & 1 & 5 & 1 & 0 & 6 & 2 \\ 1 & 1 & 2 & 1 & 6 & 5 & 2 & 3 \\ 0 & 3 & 4 & 1 & 6 & 5 & 6 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 4 \\ 3 \\ 6 \\ 5 \\ 4 \\ 2 \end{pmatrix}$$

and the central map $\mathcal{F} : \mathbb{F}^8 \to \mathbb{F}^3$ is given by polynomials

$$\begin{aligned} f^{(1)} = \; & x_1^2 + x_1x_2 + 3x_1x_3 + 2x_1x_4 + 6x_1x_5 + 4x_1x_6 + 2x_1x_7 + 5x_1x_8 + 3x_2^2 + x_2x_3 + 5x_2x_4 \\ & + x_2x_6 + 3x_2x_7 + 3x_2x_8 + 4x_3x_4 + 6x_3x_6 + 2x_3x_7 + 5x_3x_8 + 4x_4x_5 + 5x_4x_6 + 3x_4x_7 \\ & + 2x_4x_8 + 3x_2 + 6x_3 + 3x_4 + 5x_5 + 6x_6 + 2x_7 + 2x_8 + 4 \\ f^{(2)} = \; & 2x_1^2 + 5x_1x_3 + 2x_1x_4 + 4x_1x_5 + 6x_1x_6 + 5x_1x_7 + 3x_1x_8 + 4x_2^2 + x_2x_3 + 2x_2x_5 + 5x_2x_6 \\ & + 4x_2x_7 + 5x_2x_8 + 2x_3^2 + 3x_3x_4 + 6x_3x_6 + x_3x_7 + 5x_3x_8 + 2x_4x_5 + x_4x_6 + x_4x_7 + 2x_4x_8 \\ & + x_1 + x_3 + 6x_4 + 2x_5 + 4x_6 + 6x_7 + 2x_8 + 6 \\ f^{(3)} = \; & f^{(1)} + f^{(2)} = 3x_1^2 + x_1x_2 + x_1x_3 + 4x_1x_4 + 3x_1x_5 + 3x_1x_6 + x_1x_8 + 2x_2x_3 + 5x_2x_4 \\ & + 2x_2x_5 + 6x_2x_6 + x_2x_8 + 2x_3^2 + 5x_3x_6 + 3x_3x_7 + 3x_3x_8 + 6x_4x_5 + 6x_4x_6 + 4x_4x_7 \\ & + 4x_4x_8 + x_1 + 3x_2 + 2x_4 + 3x_6 + x_7 + 4x_8 + 3 \\ f^{(4)} = \; & f^{(2)} + f^{(3)} = 5x_1^2 + x_1x_2 + 6x_1x_3 + 6x_1x_4 + 2x_1x_6 + 5x_1x_7 + 4x_1x_8 + 4x_2^2 + 3x_2x_3 \\ & + 5x_2x_4 + 4x_2x_5 + 4x_2x_6 + 4x_2x_7 + 6x_2x_8 + 4x_3^2 + 3x_3x_4 + 4x_3x_6 + 4x_3x_7 + x_3x_8 \\ & + x_4x_5 + 5x_4x_7 + 6x_4x_8 + 2x_1 + 3x_2 + x_3 + x_4 + 2x_5 + 6x_8 + 2 \end{aligned}$$

We compute the public key $\mathcal{P} = (p^{(1)}, p^{(2)}, p^{(3)}) : \mathbb{F}^8 \to \mathbb{F}^3$ by $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$, which results in

$$\begin{aligned} p^{(1)} = \; & 3x_1x_2 + 2x_1x_3 + 5x_1x_4 + 6x_1x_5 + x_1x_6 + 5x_1x_7 + 6x_1x_8 + 3x_2^2 + 6x_2x_3 + 2x_2x_4 + 6x_2x_5 \\ & + 6x_3^2 + 5x_3x_4 + x_3x_5 + 6x_3x_6 + 6x_3x_8 + 6x_4^2 + 4x_4x_6 + 6x_4x_8 + 5x_5^2 + 2x_5x_6 + 4x_5x_7 \\ & + 2x_5x_8 + 2x_6^2 + x_6x_7 + 5x_6x_8 + 2x_7^2 + 3x_7x_8 + 2x_8^2 + 3x_1 + 4x_2 + 2x_3 + x_4 + 4x_5 + 3x_6 \\ & + 4x_7 + 2x_8 \\ p^{(2)} = \; & 2x_1^2 + 4x_1x_2 + x_1x_3 + 2x_1x_4 + 2x_1x_7 + x_1x_8 + x_2^2 + 5x_2x_3 + 6x_2x_4 + 2x_2x_5 + 2x_2x_6 + x_2x_7 \\ & + 2x_2x_8 + 6x_3x_4 + 6x_3x_5 + 2x_3x_6 + 2x_3x_7 + x_3x_8 + 5x_4x_6 + 3x_4x_7 + x_4x_8 + 3x_5^2 + 3x_5x_6 \\ & + 3x_5x_8 + 6x_6x_7 + 6x_6x_8 + x_7^2 + x_7x_8 + 2x_8^2 + 3x_1 + x_3 + 3x_4 + 3x_5 + x_6 + 6x_8 + 5 \\ p^{(3)} = \; & 2x_1^2 + 3x_1x_3 + 6x_1x_5 + x_1x_6 + 4x_2^2 + 4x_2x_3 + x_2x_4 + x_2x_5 + 2x_2x_6 + x_2x_7 + 2x_2x_8 + 6x_3^2 \\ & + 4x_3x_4 + x_3x_6 + 2x_3x_7 + 6x_4^2 + 2x_4x_6 + 3x_4x_7 + x_5^2 + 5x_5x_6 + 4x_5x_7 + 5x_5x_8 + 2x_6^2 \\ & + 4x_6x_8 + 3x_7^2 + 4x_7x_8 + 4x_8^2 + 6x_1 + 4x_2 + 3x_3 + 4x_4 + 4x_6 + 4x_7 + x_8 + 5 \\ p^{(4)} = \; & 4x_1^2 + 4x_1x_2 + 4x_1x_3 + 2x_1x_4 + 6x_1x_5 + x_1x_6 + 2x_1x_7 + x_1x_8 + 5x_2^2 + 2x_2x_3 + 3x_2x_5 \\ & + 4x_2x_6 + 2x_2x_7 + 4x_2x_8 + 6x_3^2 + 3x_3x_4 + 6x_3x_5 + 3x_3x_6 + 4x_3x_7 + x_3x_8 + 6x_4^2 + 6x_4x_7 \\ & + x_4x_8 + 4x_5^2 + x_5x_6 + 4x_5x_7 + x_5x_8 + 2x_6^2 + 6x_6x_7 + 3x_6x_8 + 4x_7^2 + 5x_7x_8 + 6x_8^2 + 2x_1 \\ & + 4x_2 + 4x_3 + 3x_5 + 5x_6 + 4x_7 + 3 \end{aligned}$$

Signature Generation: In order to generate a signature for the message $\mathbf{w} = (2, 2, 4, 6)$, we first need to compute $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{w})$. We choose random values for the Vinegar variables

$(x_1, x_2, x_3, x_4, x_5) = (5, 1, 6, 1, 1)$ *and substitute them into the polynomials* $f^{(1)}, f^{(2)}, f^{(3)}$ *and* $f^{(4)}$*. Thus, we obtain a linear system in the Oil variables* x_6, x_7 *and* x_8 *of the form*

$$\begin{aligned}
\bar{f}^{(1)} &= 5x_6 + 2x_7 + 6x_8 + 1\\
\bar{f}^{(2)} &= 6x_6 + 5x_8 + 2\\
\bar{f}^{(3)} &= 4x_6 + 2x_7 + 4x_8 + 3\\
\bar{f}^{(3)} &= 3x_6 + 2x_7 + 3x_8 + 5.
\end{aligned}$$

By Gaussian elimination, this system has the solution $(x_6, x_7, x_8) = (1, 3, 3)$*. Attaching the Vinegar variables yields*

$$\mathbf{y} = \mathcal{F}^{-1}(\mathbf{w}) = (5, 1, 6, 1, 1, 1, 3, 3).$$

Finally, we compute

$$\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y}) = (4, 5, 6, 0, 1, 0, 4, 2)$$

to obtain a signature $\mathbf{z} \in \mathbb{F}^8$ *for the message* $\mathbf{w}$*.*

Signature Verification: In order to check if $\mathbf{z}$ *is indeed a valid signature for the message* $\mathbf{w}$*, we compute*

$$\mathbf{w}' = \mathcal{P}(\mathbf{z}) = (2, 2, 4, 6).$$

Since $\mathbf{w}' = \mathbf{w}$ *holds, the signature is accepted.*

5.2.1. A Weakened DSFM2 UOV Signature Scheme Forgery Methodology

The algorithm to forge the signature of a weak UOV scheme by DSFM2 is described in Algorithm 14 as below.

Algorithm 14 Forgery of Weakened DSFM2 UOV Signature Scheme

```
Input: Public key $\mathcal{P}$, document $d$
Output: Signature $\mathbf{z}'$ such that $\mathcal{P}(\mathbf{z})' = \mathbf{w}' = \mathbf{w}$
1.   Solve $p^{(i)}(x) = 0$ and $p^{(k)}(x) = 0$ where $p^{(j)} = p^{(i)} + p^{(k)}$, and obtain $\mathbf{z}' = z_1, \ldots, z_o$.
```

Since $p^{(j)} = p^{(i)} + p^{(k)}$, solving the two polynomials $p^{(i)}$ and $p^{(k)}$ would solve the whole system $\mathcal{P}$.

In the following example, we show how an impersonator successfully forge the signature of a weakened DSFM2 UOV scheme.

Example 5. *Given the public key* $\mathcal{P} = (p^{(1)}, p^{(2)}, p^{(3)}, p(4))$ *of a weakened DSFM2 UOV scheme as in Example 4:*

$$\begin{aligned}
p^{(1)} = {} & 3x_1x_2 + 2x_1x_3 + 5x_1x_4 + 6x_1x_5 + x_1x_6 + 5x_1x_7 + 6x_1x_8 + 3x_2^2 + 6x_2x_3 + 2x_2x_4 + 6x_2x_5\\
& + 6x_3^2 + 5x_3x_4 + x_3x_5 + 6x_3x_6 + 6x_3x_8 + 6x_4^2 + 4x_4x_6 + 6x_4x_8 + 5x_5^2 + 2x_5x_6 + 4x_5x_7\\
& + 2x_5x_8 + 2x_6^2 + x_6x_7 + 5x_6x_8 + 2x_7^2 + 3x_7x_8 + 2x_8^2 + 3x_1 + 4x_2 + 2x_3 + x_4 + 4x_5 + 3x_6\\
& + 4x_7 + 2x_8\\
p^{(2)} = {} & 2x_1^2 + 4x_1x_2 + x_1x_3 + 2x_1x_4 + 2x_1x_7 + x_1x_8 + x_2^2 + 5x_2x_3 + 6x_2x_4 + 2x_2x_5 + 2x_2x_6 + x_2x_7\\
& + 2x_2x_8 + 6x_3x_4 + 6x_3x_5 + 2x_3x_6 + 2x_3x_7 + x_3x_8 + 5x_4x_6 + 3x_4x_7 + x_4x_8 + 3x_5^2 + 3x_5x_6\\
& + 3x_5x_8 + 6x_6x_7 + 6x_6x_8 + x_7^2 + x_7x_8 + 2x_8^2 + 3x_1 + x_3 + 3x_4 + 3x_5 + x_6 + 6x_8 + 5
\end{aligned}$$

$$\begin{aligned}
p^{(3)} = {} & 2x_1^2 + 3x_1x_3 + 6x_1x_5 + x_1x_6 + 4x_2^2 + 4x_2x_3 + x_2x_4 + x_2x_5 + 2x_2x_6 + x_2x_7 + 2x_2x_8 + 6x_3^2 \\
& + 4x_3x_4 + x_3x_6 + 2x_3x_7 + 6x_4^2 + 2x_4x_6 + 3x_4x_7 + x_5^2 + 5x_5x_6 + 4x_5x_7 + 5x_5x_8 + 2x_6^2 \\
& + 4x_6x_8 + 3x_7^2 + 4x_7x_8 + 4x_8^2 + 6x_1 + 4x_2 + 3x_3 + 4x_4 + 4x_6 + 4x_7 + x_8 + 5 \\
p^{(4)} = {} & 4x_1^2 + 4x_1x_2 + 4x_1x_3 + 2x_1x_4 + 6x_1x_5 + x_1x_6 + 2x_1x_7 + x_1x_8 + 5x_2^2 + 2x_2x_3 + 3x_2x_5 \\
& + 4x_2x_6 + 2x_2x_7 + 4x_2x_8 + 6x_3^2 + 3x_3x_4 + 6x_3x_5 + 3x_3x_6 + 4x_3x_7 + x_3x_8 + 6x_4^2 + 6x_4x_7 \\
& + x_4x_8 + 4x_5^2 + x_5x_6 + 4x_5x_7 + x_5x_8 + 2x_6^2 + 6x_6x_7 + 3x_6x_8 + 4x_7^2 + 5x_7x_8 + 6x_8^2 + 2x_1 \\
& + 4x_2 + 4x_3 + 3x_5 + 5x_6 + 4x_7 + 3
\end{aligned}$$

Let $\mathbf{w}=(2,2,4,6)$. *The impersonator computes* $\bar{p}^{(i)} = p^{(i)} - w_i \pmod 7$ *where* $i = 1,2,3,4$. *Then, to solve* $\bar{p}^{(1)}(x) = 0$, *he chooses random variables* $(x_1,x_2,x_3,x_4,x_5,x_6,x_7) = (5,6,0,5,4,7,2)$ *and substitutes them into* $\bar{p}^{(1)}$ *and* $\bar{p}^{(2)}$ *which results quadratic equations with one variable*

$$\begin{aligned}
2x_8^2 + 6x_8 + 4 = 0 \\
2x_8^2 + 6 = 0.
\end{aligned}$$

Since these equations has a solution $x_8 = 5$ *hence the solution for* $p^{(1)}$ *and* $p^{(2)}$ *are* $(5,6,0,5,4,7,2,5)$. *This solution is also the solution to* $p^{(3)}$ *and* $p^{(4)}$, *which implies* $\mathcal{P}(\mathbf{z}') = 0$ *where* $\mathbf{z}' = (5,6,0,5,4,7,2,5)$ *is the forged signature. Indeed* $\mathbf{z}' \neq \mathbf{z} = (4,5,6,0,1,0,4,2)$.

5.2.2. Identifying a Weakened DSFM2 UOV Scheme

We can directly use Algorithm 5 to identify a weakened DSFM2 UOV scheme.

Example 6. *Given the public key* $\mathcal{P} = (p^{(1)}, p^{(2)}, p^{(3)}, p^{(4)})$ *of a weakened DSFM2 UOV scheme as in Example 4:*

$$\begin{aligned}
p^{(1)} = {} & 3x_1x_2 + 2x_1x_3 + 5x_1x_4 + 6x_1x_5 + x_1x_6 + 5x_1x_7 + 6x_1x_8 + 3x_2^2 + 6x_2x_3 + 2x_2x_4 + 6x_2x_5 \\
& + 6x_3^2 + 5x_3x_4 + x_3x_5 + 6x_3x_6 + 6x_3x_8 + 6x_4^2 + 4x_4x_6 + 6x_4x_8 + 5x_5^2 + 2x_5x_6 + 4x_5x_7 \\
& + 2x_5x_8 + 2x_6^2 + x_6x_7 + 5x_6x_8 + 2x_7^2 + 3x_7x_8 + 2x_8^2 + 3x_1 + 4x_2 + 2x_3 + x_4 + 4x_5 + 3x_6 \\
& + 4x_7 + 2x_8 \\
p^{(2)} = {} & 2x_1^2 + 4x_1x_2 + x_1x_3 + 2x_1x_4 + 2x_1x_7 + x_1x_8 + x_2^2 + 5x_2x_3 + 6x_2x_4 + 2x_2x_5 + 2x_2x_6 + x_2x_7 \\
& + 2x_2x_8 + 6x_3x_4 + 6x_3x_5 + 2x_3x_6 + 2x_3x_7 + x_3x_8 + 5x_4x_6 + 3x_4x_7 + x_4x_8 + 3x_5^2 + 3x_5x_6 \\
& + 3x_5x_8 + 6x_6x_7 + 6x_6x_8 + x_7^2 + x_7x_8 + 2x_8^2 + 3x_1 + x_3 + 3x_4 + 3x_5 + x_6 + 6x_8 + 5 \\
p^{(3)} = {} & 2x_1^2 + 3x_1x_3 + 6x_1x_5 + x_1x_6 + 4x_2^2 + 4x_2x_3 + x_2x_4 + x_2x_5 + 2x_2x_6 + x_2x_7 + 2x_2x_8 + 6x_3^2 \\
& + 4x_3x_4 + x_3x_6 + 2x_3x_7 + 6x_4^2 + 2x_4x_6 + 3x_4x_7 + x_5^2 + 5x_5x_6 + 4x_5x_7 + 5x_5x_8 + 2x_6^2 \\
& + 4x_6x_8 + 3x_7^2 + 4x_7x_8 + 4x_8^2 + 6x_1 + 4x_2 + 3x_3 + 4x_4 + 4x_6 + 4x_7 + x_8 + 5 \\
p^{(4)} = {} & 4x_1^2 + 4x_1x_2 + 4x_1x_3 + 2x_1x_4 + 6x_1x_5 + x_1x_6 + 2x_1x_7 + x_1x_8 + 5x_2^2 + 2x_2x_3 + 3x_2x_5 \\
& + 4x_2x_6 + 2x_2x_7 + 4x_2x_8 + 6x_3^2 + 3x_3x_4 + 6x_3x_5 + 3x_3x_6 + 4x_3x_7 + x_3x_8 + 6x_4^2 + 6x_4x_7 \\
& + x_4x_8 + 4x_5^2 + x_5x_6 + 4x_5x_7 + x_5x_8 + 2x_6^2 + 6x_6x_7 + 3x_6x_8 + 4x_7^2 + 5x_7x_8 + 6x_8^2 + 2x_1 \\
& + 4x_2 + 4x_3 + 3x_5 + 5x_6 + 4x_7 + 3
\end{aligned}$$

To identify $\mathcal{P}$ *is a forgeable system, we take two polynomials* $p^{(1)}$ *and* $p^{(2)}$ *and compute*

$$\begin{aligned}
p^{(1)}+p^{(2)}\ (\mathrm{mod}\ 7) = {} & 2x_1^2+4x_1x_2+x_1x_3+2x_1x_4+2x_1x_7+x_1x_8+x_2^2+5x_2x_3+6x_2x_4+2x_2x_5 \\
& +2x_2x_6+x_2x_7+2x_2x_8+6x_3x_4+6x_3x_5+2x_3x_6+2x_3x_7+x_3x_8+5x_4x_6 \\
& +3x_4x_7+x_4x_8+3x_5^2+3x_5x_6+3x_5x_8+6x_6x_7+6x_6x_8+x_7^2+x_7x_8+2x_8^2 \\
& +3x_1+x_3+3x_4+3x_5+x_6+6x_8+5 \\
p^{(2)}+p^{(3)}\ (\mathrm{mod}\ 7) = {} & 4x_1^2+4x_1x_2+4x_1x_3+2x_1x_4+6x_1x_5+x_1x_6+2x_1x_7+x_1x_8+5x_2^2+2x_2x_3 \\
& +3x_2x_5+4x_2x_6+2x_2x_7+4x_2x_8+6x_3^2+3x_3x_4+6x_3x_5+3x_3x_6+4x_3x_7 \\
& +x_3x_8+6x_4^2+6x_4x_7+x_4x_8+4x_5^2+x_5x_6+4x_5x_7+x_5x_8+2x_6^2+6x_6x_7 \\
& +3x_6x_8+4x_7^2+5x_7x_8+6x_8^2+2x_1+4x_2+4x_3+3x_5+5x_6+4x_7+3
\end{aligned}$$

Since

$$\begin{aligned}
p^{(1)}+p^{(2)} &= p^{(3)}\ (\mathrm{mod}\ 7) \\
p^{(2)}+p^{(3)} &= p^{(4)}\ (\mathrm{mod}\ 7)
\end{aligned}$$

we have successfully identified that $\mathcal{P}$ is a forgeable system.

6. The Inability to Generate Weak Rainbow Signature Scheme via DSFM1 and DSFM2 Methodologies

Firstly, we observe that the central map $\mathcal{F}$ of a UOV scheme in the form of

$$f^{(k)} = \sum_{a,b\in V} \alpha_{a,b}^{(k)} x_a x_b + \sum_{a\in V, b\in O} \beta_{a,b}^{(k)} x_a x_b + \sum_{a\in V\cup O} \gamma_a^{(k)} x_a + \delta^{(k)} (k=1,\ldots,o).$$

Thus, all polynomials $f^{(j)} = k_j f^{(1)}$ where $j = 2,\ldots,m$ and $f^{(j)} = f^{(i)} + f^{(k)}$ where $j = 2,\ldots,m$ and $i,k = 1,\ldots,j-1$ in the central map $\mathcal{F}$ are of the same form as above.

Thus, the inability to generate a weak Rainbow signature scheme via DSFM1 and DSFM2 methodologies is because of its central map $\mathcal{F}$ having the form of

$$f^{(k)}(\mathbf{x}) = \sum_{a,b\in V_\ell, a\le b} \alpha_{a,b}^{(k)} x_a x_b + \sum_{a\in O_\ell, b\in V_\ell} \beta_{a,b}^{(k)} x_a x_b + \sum_{a\in V_\ell\cup O_\ell} \gamma_a^{(k)} x_a + \eta^{(k)} (k = v_1+1,\ldots,n).$$

The form of polynomials $f^{(k)}(\mathbf{x})$ are different depending on the ℓ-th level. As we can see, the index i and j for the variables are from the index sets V_ℓ and O_ℓ where ℓ is the only integer such that $k \in O_\ell$. For instance, when $\ell = 1$, we will have $O_1 = \{v_1+1,\ldots,v_2\}$ and $V_1 = \{1,\ldots v_1\}$. The value of k is taken from the set O_1. Therefore, the polynomials $f^{(v_1+1)}(\mathbf{x}),\ldots,f^{(v_2)}(\mathbf{x})$ will share the same form. For $\ell = 2$, $O_2 = \{v_2+1,\ldots v_3\}$ and $V_2 = \{1,\ldots,v_2\}$ where $k \in O_2$, the polynomials $f^{(v_2+1)}(\mathbf{x}),\ldots,f^{(v_3)}(\mathbf{x})$ are of the same form. Since the polynomials have different variable forms, we cannot construct the central map $\mathcal{F}$ as in Algorithm 2 and the polynomials in $\mathcal{P}$ of the form $p^{(j)} = p^{(i)} + p^{(k)}$ as in Algorithm 4.

7. Generating Weak UOV and Rainbow Signature Scheme

In the following example, we illustrate the generation of weak UOV and Rainbow schemes via DSFM3 methodology as well as the signing and verification process. Firstly, the public–private key pair of either UOV or Rainbow is generated as in the original version of the schemes. Secondly, suppose the RCA computes $\mathbf{x} = (x_1,\ldots,x_m)$ such that $\mathcal{P}(\mathbf{x}) = 0$ and shares the vector $\mathbf{x}$ with the adversary. The adversary can forge the signature $\mathbf{x}$ via DSFM3. The example below shows that weak UOV and Rainbow schemes can still be used by a user without suspicion since the constants seem randomized and the signing and verification work as normal.

Example 7. *Let $\mathcal{P} = (p^{(1)}, p^{(2)}, p^{(3)}) : \mathbb{F}^7 \to \mathbb{F}^3$ be a valid public key over $\mathbb{F} = GF(53)$ that can be utilized for both UOV and Rainbow signature schemes. Suppose $\mathbf{x} = (35, 46, 24, 57, 21, 27, 25)$ such that $\mathcal{P}(\mathrm{x}) = 0$. The adversary is given the integer set* **x** *from the RCA and suppose the adversary wants to forge the signature* $\mathbf{z} = (40, 46, 24, 57, 21, 3, 34)$ *corresponding to* $\boldsymbol{w} = (1, 30, 46)$.

$$\begin{aligned}
p^{(1)} = \; & 47x_1^2 + 33x_1x_2 + 22x_1x_3 + 38x_1x_4 + 45x_1x_5 + 17x_1x_6 + 18x_1x_7 + 8x_2^2 + 23x_2x_3 + 18x_2x_4 \\
& + x_2x_5 + 44x_2x_6 + 41x_2x_7 + 12x_3^2 + 47x_3x_4 + 6x_3x_5 + 15x_3x_6 + 11x_3x_7 + 22x_4^2 + x_4x_5 \\
& + 23x_4x_6 + 13x_4x_7 + 9x_5^2 + 39x_5x_6 + 42x_5x_7 + 30x_5 + 15x_6^2 + 48x_6x_7 + 11x_7^2 + 50x_1 \\
& + 13x_2 + 11x_3 + 5x_4 + 30x_5 + x_6 + 50x_7 + 32 \\
p^{(2)} = \; & 45x_1^2 + 6x_1x_3 + 12x_1x_4 + 16x_1x_5 + 26x_1x_6 + 46x_1x_7 + 3x_1x_2 + 13x_2^2 + 30x_2x_3 + 47x_2x_4 \\
& + 43x_2x_5 + 14x_2x_6 + 30x_2x_7 + 39x_3^2 + 17x_3x_4 + 15x_3x_5 + 46x_3x_6 + 40x_3x_7 + 45x_4^2 \\
& + 18x_4x_5 + 22x_4x_6 + 9x_4x_7 + 3x_5^2 + 37x_5x_6 + 35x_5x_7 + 14x_6^2 + 38x_6x_7 + 26x_7^2 + 46x_1 \\
& + 37x_2 + 37x_3 + 44x_4 + 28x_5 + 12x_6 + 10x_7 + 35 \\
p^{(3)} = \; & 21x_1^2 + 13x_1x2 + 26x_1x_3 + 14x_1x_4 + 44x_1x_5 + 12x_1x_7 + 37x_2^2 + 18x_2x_3 + 18x_2x_4 + 49x_2x_5 \\
& + 4x_2x_6 + 29x_2x_7 + 11x_3^2 + 14x_3x_4 + 22x_3x_5 + 27x_3x_6 + 13x_3x_7 + 2x_4^2 + 30x_4x_5 + 4x_4x_6 \\
& + 14x_4x_7 + 45x_5^2 + 39x_5x_6 + x_5x_7 + 2x_6^2 + 49x_6x_7 + 24x_7^2 + 20x_1 + 32x_2 + 30x_3 + 34x_4 \\
& + 43x_5 + 32x_6 + 30x_7 + 40
\end{aligned}$$

The adversary computes $\mathcal{P}(\mathbf{x} + \alpha)$ and obtains

$$\begin{aligned}
p^{(1)} &= 33\alpha^2 + 13\alpha \\
p^{(2)} &= 46\alpha^2 + \alpha \\
p^{(3)} &= 52\alpha^2 + 25\alpha.
\end{aligned}$$

Solving $p^{(1)} = w_1$, $p^{(2)} = w_2$ and $p^{(3)} = w_3$, the adversary will obtain $\alpha = 23$. Therefore, $\mathbf{z}' = (5, 16, 47, 27, 44, 50, 48)$ is the forged signature. Indeed $\mathbf{z}' \neq \mathbf{z} = (40, 46, 24, 57, 21, 3, 34)$. The verification process will be successful since:

$$\begin{aligned}
p^{(1)}(5, 16, 47, 27, 44, 50, 48) &= 0 \\
p^{(2)}(5, 16, 47, 27, 44, 50, 48) &= 0 \\
p^{(3)}(5, 16, 47, 27, 44, 50, 48) &= 0.
\end{aligned}$$

8. Discussion

Our work enabled us to showcase the practicality of the DSFM1, DSFM2 and DSFM3 methodologies to forge UOV and Rainbow signatures. The strategies outlined to identify whether DSFM1 or DSFM2 was applied on UOV and Rainbow parameters must be adhered to in order to ensure the security of the signature. As discussed on [29], the complexity to conduct due diligence are $O(m)$ and $O(m^3)$, respectively, where m is the number of equations. However, to this end, it is still unanswered whether there are possible mechanisms to identify DSFM3 weakened systems. The DSFM3 is deployed on random polynomials, and does not involve modification on polynomials to make it vulnerable. As such, the system $\mathcal{P}$ has no anomalies. Instead, the adversary only needs to solicit the vector **x** which satisfies $\mathcal{P}(\mathbf{x}) = 0$ from the RCA.

9. Conclusions

In conclusion, we have revisited two signature forgery methodologies (DSFM1 and DSFM2) and put forward one novel signature forgery methodology, DSFM3. The public key system $\mathcal{P}$ of a UOV signature scheme is not secure if it is generated using DSFM1, DSFM2 and DSFM3 methodologies by RCA. Potential users of the UOV signature scheme are able to identify whether the public parameters are generated via DSFM1 and DSFM2 methodologies. As such they must conduct due diligence upon receiving the public key system $\mathcal{P}$. To this end, the Rainbow signature scheme is resistant to DSFM1 as well as DSFM2 methodologies and is only vulnerable to the DSFM3 methodology. However, it is still an open question whether a public key system $\mathcal{P}$ of UOV and Rainbow signature schemes can be identified if it is generated via DSFM3 methodology since there are no anomalies in the public key.

Author Contributions: Conceptualization, N.A.S.A.J., M.R.K.A., S.H.S. and K.A.; Formal analysis, N.A.S.A.J. and M.R.K.A.; Funding acquisition, M.R.K.A.; Investigation, N.A.S.A.J., M.R.K.A., S.H.S. and K.A. Methodology, N.A.S.A.J. and M.R.K.A.; Project administration, M.R.K.A.; Supervision, M.R.K.A. and S.H.S.; Validation, M.R.K.A.; Visualization, N.A.S.A.J., M.R.K.A., S.H.S. and K.A.; Writing—original draft, N.A.S.A.J.; Writing—review & editing, M.R.K.A. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported by Ministry of Education of Malaysia with Fundamental Research Grant Scheme (FRGS/1/2019/STG06/UPM/02/8). It is also partially supported by Mediterranea Universiti of Reggio Calabria (UNIRC) Research Grant (UPM/INSPEM/700-3/1/GERANANTARABAN GSA/6380 073122-10065).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The first author would like to further express appreciation to the Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM), and Ministry of Higher Education (MOHE) for giving the opportunity to conduct this research.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CA	Certificate Authority
DSFM1	Digital Signature Forgery Mechanism 1
DSFM2	Digital Signature Forgery Mechanism 2
DSFM3	Digital Signature Forgery Mechanism 3
DLP	Discrete Logarithm Problem
IFP	Integer Factorization Problem
MQP	Multivariate Quadratic Problem
RCA	Rogue Certificate Authority
RSA	Rivest-Shamir-Adleman
UOV	Unbalance Oil and Vinegar

References

1. Dong, Z.; Kane, K.; Camp, L.J. Detection of Rogue Certificates from Trusted Certificate Authorities Using Deep Neural Networks. *ACM Trans. Priv. Secur.* **2016**, *19*, 1–31. [CrossRef]
2. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
3. Rivest, R.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
4. Diffie, W.; Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
5. Ding, J.; Petzoldt, A. Current state of multivariate cryptography. *IEEE Secur. Priv.* **2017**, *15*, 28–36. [CrossRef]
6. Kipnis, A.; Patarin, J.; Goubin, L. Unbalanced oil and vinegar signature schemes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 206–222.
7. Patarin, J. The oil and vinegar signature scheme. In Proceedings of the Dagstuhl Workshop on Cryptography, Saarbrucken, Germany, 22–26 September 1997.

8. Kipnis, A.; Shamir, A. Cryptanalysis of the oil and vinegar signature scheme. In Proceedings of the 18th Annual International Cryptology Conference, Santa Barbara, CA, USA, 23–27 August 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 257–266.
9. Ding, J.; Schmidt, D. Rainbow, a new multivariable polynomial signature scheme. In Proceedings of the International Conference on Applied Cryptography and Network Security, New York, NY, USA, 7–10 June 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 164–175.
10. Beullens, W.; Preneel, B. Field lifting for smaller UOV public keys. In Proceedings of the International Conference on Cryptology in India, Chennai, India, 10–13 December 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 227–246.
11. Petzoldt, A. Efficient key generation for rainbow. In Proceedings of the International Conference on Post-Quantum Cryptography, Paris, France, 15–17 April 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 92–107.
12. Li, J.; Hu, Z.; Kais, S. Practical quantum encryption protocol with varying encryption configurations. *Phys. Rev. Res.* **2021**, *3*, 023251. [CrossRef]
13. Feng, Y.; Zhou, J.; Li, J.; Zhao, W.; Shi, J.; Shi, R.; Li, W. SKC-CCCO: an encryption algorithm for quantum group signature. *Quantum Inf. Process.* **2022**, *21*, 1–29. [CrossRef]
14. Shi, J.; Lu, Y.; Feng, Y.; Huang, D.; Lou, X.; Li, Q.; Shi, R. A quantum hash function with grouped coarse-grained boson sampling. *Quantum Inf. Process.* **2022**, *21*, 1–17. [CrossRef]
15. Shi, J.; Chen, S.; Lu, Y.; Feng, Y.; Shi, R.; Yang, Y.; Li, J. An approach to cryptography based on continuous-variable quantum neural network. *Sci. Rep.* **2020**, *10*, 2107. [CrossRef] [PubMed]
16. Feng, Y.; Shi, R.; Shi, J.; Zhao, W.; Lu, Y.; Tang, Y. Arbitrated quantum signature protocol with boson sampling-based random unitary encryption. *J. Phys. A Math. Theor.* **2020**, *53*, 135301. [CrossRef]
17. Lyubashevsky, V.; Ducas, L.; Kiltz, E.; Lepoint, T.; Schwabe, P.; Seiler, G.; Stehlé, D.; Avanzi, R.; Bos, J.; Schanck, J. CRYSTALS-Dilithium; *Submiss. NIST Post-Quantum Cryptogr. Stand.* **2017**, 1–29. (MDPI: Please add volume number and confirm type of this reference.)
18. Fouque, P.A;, Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. *Submiss. NIST Post-Quantum Cryptogr. Stand. Process.* **2018**, *36*, 1–75.
19. Bernstein, D.J.; Chou, T.; Lange, T.; von Maurich, I.; Misoczki, R.; Niederhagen, R.; Persichetti, E.; Peters, C.; Schwabe, P.; Sendrier, N.; et al. Classic McEliece: Conservative code-based cryptography. In Proceedings of the PQCRYPTO Mini-School and Workshop, Taipei, Taiwan, 27–29 June 2018.
20. Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. In Proceedings of the 2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK, 24–26 April 2018; pp. 353–367.
21. Chen, C.; Danba, O.; Hoffstein, J.; Hülsing, A.; Rijneveld, J.; Schanck, J.M.; Schwabe, P.; Whyte, W.; Zhang, Z. *Algorithm Specifications and Supporting Documentation*; Brown University: Providence, RI, USA; Onboard Security Company: Wilmington, DE, USA, 2019.
22. D'Anvers, J.P.; Karmakar, A.; Sinha Roy, S.; Vercauteren, F. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In Proceedings of the International Conference on Cryptology in Africa, Marrakesh, Morocco, 7–9 May 2018; Springer: Cham, Switzerland, 2018; pp. 282–305.
23. Beullens, W. Improved cryptanalysis of UOV and rainbow. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 17–21 October 2021; Springer: Cham, Switzerland, 2021; pp. 348–373.
24. Beullens, W. Breaking rainbow takes a weekend on a laptop. *Cryptol. ePrint Arch.* **2022**, 214.
25. Cartor, R.; Cartor, M.; Lewis, M.; Smith-Tone, D. IPRainbow. In Proceedings of the International Conference on Post-Quantum Cryptography, Virtual, 28–30 September 2022; Springer: Cham, Switzerland, 2022; pp. 170–184.
26. Thomae, E.; Wolf, C. Cryptanalysis of enhanced TTS, STS and all its variants, or: Why cross-terms are important. In Proceedings of the International Conference on Cryptology in Africa, Ifrane, Morocco, 10–12 July 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 188–202.
27. Chen, J.; Ning, J.; Ling, J.; Lau, T. S. C.; Wang, Y. A new encryption scheme for multivariate quadratic systems. *Theor. Comput. Sci.* **2020**, *809*, 372–383. [CrossRef]
28. Chakraborty, O.; Faugére, J. C.; Perret, L. Cryptanalysis of the extension field cancellation cryptosystem. *Des. Codes Cryptogr.* **2021**, *89*, 1335–1364. [CrossRef]
29. Jamal, N.A.S.A.; Ariffin, M.R.K.; Sapar, S.H.; Abdullah, K. *Novel Forgery Mechanisms in Multivariate Signature Schemes*; Institute for Mathematical Research, Universiti Putra Malaysia: Serdang, Malaysia, 2022.