Fault Attacks on UOV and Rainbow

Juliane Krämer and Mirjam Loiero

TU Darmstadt, Germany
jkraemer@cdc.informatik.tu-darmstadt.de

Abstract. Multivariate cryptography is one of the main candidates for creating post-quantum public key cryptosystems. Especially in the area of digital signatures, there exist many practical and secure multivariate schemes. The signature schemes UOV and Rainbow are two of the most promising and best studied multivariate schemes which have proven secure for more than a decade. However, so far the security of multivariate signature schemes towards physical attacks has not been appropriately assessed. Towards a better understanding of the physical security of multivariate signature schemes, this paper presents fault attacks against SingleField schemes, especially UOV and Rainbow. Our analysis shows that although promising attack vectors exist, multivariate signature schemes inherently offer a good protection against fault attacks.

Keywords: Multivariate cryptography · Rainbow · UOV · Fault Attacks

1 Introduction

Cryptographic techniques are an essential tool to guarantee the security of communication in modern society. Until recently, the security of nearly all cryptographic schemes used in practice was based on number theoretic problems such as factoring large integers and solving discrete logarithms, e.g., RSA and ECC. However, schemes like these will become insecure once large enough quantum computers are built. This is due to Shor's algorithm [21], which solves the integer factorization problem and discrete logarithms in polynomial time on a quantum computer. Therefore, we need alternative public key schemes which are based on hard mathematical problems that remain hard in the presence of quantum computers: post-quantum cryptosystems.

Besides cryptography based on lattices, hash functions, codes, and isogenies, multivariate cryptography is one of the main candidates for this. The security of multivariate schemes is based on the hardness of the MQ-problem - solving a randomly generated system of multivariate quadratic polynomial equations over finite fields - which is NP-hard [13]. Depending on the size of the finite field, a distinction is made between SingleField schemes and BigField schemes [20]. The public key of multivariate schemes is a set of multivariate polynomials and the private key is mainly the trapdoor that allows to invert the public key. Unfortunately, most of the proposed multivariate encryption schemes have been broken. This is due to the fact that the construction in this case must be based on an injective trapdoor function. As a consequence, the multivariate system of the public

key is not a hard instance of the MQ-problem. On the other hand, constructions of multivariate signature schemes allow to add some randomness to the secret trapdoor which leads to a harder public key. Multivariate signature schemes are in general very fast and require only modest computational ressources, which makes them attractive for the use on low cost devices like smart cards and RFID chips [4, 8]. Therefore, developing fast and light-weight implementations of multivariate signature schemes became an active field of research [9, 23, 25]. Among many practical alternatives, UOV [16] and Rainbow[1] [11] are two of the oldest, most efficient, and most promising multivariate signature schemes.

When it comes to implementing post-quantum cryptography and using it in practical applications, however, relying only on the mathematical security of the schemes is not sufficient, but the physical security of the schemes and their implementations has to be ensured as well. Since post-quantum cryptography is only rarely used in practice as of 2019, and especially not in widespread use on smart cards and in embedded systems so far, research about side channel attacks and fault attacks on these schemes is still in the early stages of development. For multivariate schemes in particular, only few publications exist, most of which target (passive) side channel attacks rather than (active) fault attacks: Already in 2001, it was theoretically shown how the signature schemes FLASH and SFLASH can be attacked with differential power analysis (DPA) [22]. Steinwaldt et al. reveal the secret 80-bit seed Δ for SHA-1 and subsequently the affine bijections $\mathcal{S}$ and $\mathcal{T}$ by analyzing the power consumption of involved $\oplus$ operations. Okeya et al. propose another side channel attack on SFLASH in 2004 [18]. They also learn Δ through a DPA and then break SFLASH by reducing its security to the C^* problem, which is broken. They verify their results experimentally. Many years later, Yi and Li present a DPA against the enTTS signature scheme [24]. The DPA attack is facilitated by a fault attack which fixes certain unknown values to known ones. The DPA part of the attack is verified experimentally against a naive ASIC implementation of enTTS. Only recently, Park et al. presented side channel attacks on the Rainbow and UOV signature schemes [19]. They use correlation power analysis together with algebraic key recovery attacks and demonstrate the practical feasibility of their attack on an 8-bit AVR microcontroller. Regarding fault attacks on multivariate cryptography, only a single work exists: Hashimoto et al. describe general methods how to attack multivariate cryptography with fault attacks [14][2]. These methods provide the basis for our work.

Our Contribution. The authors of [14] focus on BigField schemes and STS-type schemes, which form a specific subclass of SingleField schemes. We complement their work by comprehensively analyzing how the attacks can be applied to SingleField schemes in general. In particular, we apply the attacks to UOV and Rainbow. We find that several special cases exist where the attacks do not work.

[1] Rainbow has been submitted to the call for post-quantum cryptography standardization by the US American National Institute of Standards and Technology (NIST) in November 2017 [10] and was selected Round 2 Candidate in January 2019 [1].

[2] The same authors published their work additionally in [15].

From these findings we deduce countermeasures to protect multivariate signature schemes against fault attacks. With this, we pave the way for future fault attack resistant (implementations of) multivariate signature schemes.

Our analysis shows that although promising attack vectors exist, the randomness induced by the vinegar variables - and in case of Rainbow also by the different layers - proves to be an inherent protection against fault attacks.

Organization. In Section 2, we introduce the mathematics of multivariate cryptosystems and summarize the work [14]. In the subsequent Sections 3 and 4, we discuss the applicability of the attacks from [14] to SingleField schemes and in particular to UOV and Rainbow. We provide success probabilities for the attacks and detect cases where the attacks do not work. We present countermeasures to protect multivariate signature schemes against such attacks in Section 5.

2 Background

First, we provide an introduction to multivariate cryptosystems in Section 2.1. Then, in Section 2.2 we give an overview about the ideas of the attacks in [14].

2.1 Multivariate Cryptosystems

The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials, see Equation 1.

$$\begin{aligned} p^{(1)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(1)}\cdot x_i x_j + \sum_{i=1}^{n} p_i^{(1)}\cdot x_i + p_0^{(1)} \\ &\vdots \\ p^{(m)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(m)}\cdot x_i x_j + \sum_{i=1}^{n} p_i^{(m)}\cdot x_i + p_0^{(m)} \end{aligned} \tag{1}$$

The security of multivariate schemes is based on the MQ problem: Given m quadratic polynomials $p^{(1)}(\mathbf{x}),\ldots,p^{(m)}(\mathbf{x})$ in n variables $x_1,\ldots,x_n$ as shown in Equation 1, find a vector $\bar{\mathbf{x}} = (\bar{x}_1,\ldots,\bar{x}_n)$ such that $p^{(1)}(\bar{\mathbf{x}}) = \ldots = p^{(m)}(\bar{\mathbf{x}}) = 0$. The MQ problem (for $m \approx n$) is proven to be NP-hard [13].

To build a public key cryptosystem on the basis of the MQ problem, one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$, the central map, where $\mathbb{F}$ is a finite field. To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine maps $\mathcal{T} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^n$. These affine maps can be written as $\mathcal{T}(y) = Ty + t$ and $\mathcal{S}(x) = Sx + s$, where $T \in \mathbb{F}^{m\times m}$ and $S \in \mathbb{F}^{n\times n}$ are linear transformations and $t \in \mathbb{F}^m$ and $s \in \mathbb{F}^n$ are constant vectors. The *public key* of the scheme is given by $\mathcal{P} = \mathcal{T}\circ\mathcal{F}\circ\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^m$. The *private key* consists of $\mathcal{T}$, $\mathcal{F}$, and $\mathcal{S}$ and thereby allows to invert the public key. [3]

[3] Due to the above construction, the security of multivariate public key schemes is not only based on the MQ-Problem, but also on the EIP-Problem (Extended Isomorphism of Polynomials) of finding the composition of $\mathcal{P}$ [20].

Signature Generation To generate a signature for a message d, the signer uses a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^m$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^m$ and computes recursively $\mathbf{x} = \mathcal{T}^{-1}(\mathbf{w}) \in \mathbb{F}^m$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{x}) \in \mathbb{F}^n$, and $\mathbf{z} = \mathcal{S}^{-1}(\mathbf{y})$. The signature of the message d is $\mathbf{z} \in \mathbb{F}^n$. Here, $\mathcal{F}^{-1}(\mathbf{x})$ means finding one (of possibly many) preimages of $\mathbf{x}$ under the central map $\mathcal{F}$.

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

In Appendix A, we describe the signature schemes Rainbow and UOV.

2.2 General Fault Attacks on Multivariate Public Key Cryptosystems

In [14] the authors propose two approaches for fault attacks on multivariate signature schemes. In both attacks, the goal of the attackers is to reveal the affine maps $\mathcal{T}$ and $\mathcal{S}$, respectively, via a linear algebra attack [16]. By a preceding fault attack, they decrease the complexity of the linear algebra attack considerably.

The goal of the first attack, which we analyze in Section 3, is to gain partial information about the affine map $\mathcal{T}$ via fault injection on the central map $\mathcal{F}$. It is assumed that the fault changes a single coefficient during signature generation. By changing an additional coefficient in each following signature generation and using message-signature pairs for random messages, the attacker deduces information about the affine map $\mathcal{T}$.

The second attack aims at the random values which are used during signature generation. If an attacker manages to fix (some of) those values for several signature generations, he can transform the affine map $\mathcal{S}$ by using several pairs of random messages and corresponding signatures to facilitate the subsequent linear algebra attack. We analyze this attack in Section 4.

3 Fault Attack on the Central Map

In this section we analyze the fault attack on the central map. We introduce the attacker model in Section 3.1 and give a detailed description how the attack is intended to work for SingleField schemes in Section 3.2. In Section 3.3, we show that UOV schemes - contrary to what is claimed in [14] - are immune to this attack. In Section 3.4 we explain how the attack can be applied to Rainbow schemes and in Section 3.5 we analyze special cases of the attack.

3.1 Attacker Model

We assume in this attack that the attacker targets the signature generation process and randomly changes a coefficient in the central map $\mathcal{F}$. He either modifies $\mathcal{F}$ directly or he attacks the public key $\mathcal{P}$ to modify $\mathcal{F}$. (For a discussion about the distinguishability of the faulty place in the latter case, we refer to [14,

Section 3.2.3].) The fault that the attacker induces is permanent. He then receives the signature of a random message, i.e., this signature is generated with a faulty central map, and applies the correct public key to it. Afterwards, he again induces a fault into the central map - hence, the central map used in the next step to generate a signature on another random message includes two faults, and so on. By comparing the random messages with the messages yielded by signing the random messages with the faulty central map and then applying the correct public key to them, the attacker gains information about the affine map $\mathcal{T}$.

In a successful attack, all faults would affect pairwise different equations of the central map $\mathcal{F}$. The attacker would need $m-1$ faults, see Section 3.2. As of 2018, we have $m = 28$ in the Rainbow scheme for $\mathbb{F} = GF(256)$ [20, Table 6.13].

3.2 Detailed Description of the Attack for SingleField Schemes

In [14], the authors describe the attack for Stepwise Triangular System (STS) schemes. Schemes of this type form a subset of the SingleField family. However, our findings show that the applicability and the success of this attack highly depend on the concrete scheme it is targeting. Therefore, we first generalize the attack to SingleField schemes, and then approach the schemes UOV and Rainbow in a more concrete way.

For each message that is to be signed, i.e., for each iteration of the attack, in case of SingleField schemes four steps have to be performed[4]. They are displayed in Algorithm 1. Since we do not know which kind of coefficient $\alpha_{ij}^{(k)}$,$\beta_{ij}^{(k)}$, $\gamma_i^{(k)}$ or $\eta^{(k)}$ - the coefficients of the quadratic and linear variables and the constant part of the central map $\mathcal{F}$, see Appendices A.2 and A.3 - is changed, we write $\xi^{(k)}$ for any of those. We denote faulty values with an apostrophe, e.g., $\xi'^{(k)}$.

Algorithm 1 One iteration of the attack on the central map

1: Change a coefficient $\xi^{(k)}$ into $\xi'^{(k)}$ to get a faulty central map $\mathcal{F}'$ out of $\mathcal{F}$. Then $\Delta\mathcal{F} = \mathcal{F}' - \mathcal{F}$.
2: Sign a randomly chosen message $h^{(l)} = (h_1^{(l)}, \ldots, h_m^{(l)})$ via the faulty central map $\mathcal{F}'$ by $z'^{(l)} := \mathcal{S}^{-1}(\mathcal{F}'^{-1}(\mathcal{T}^{-1}(h^{(l)})))$, where $z'^{(l)} = (z'^{(l)}_1, \ldots, z'^{(l)}_n)$.
3: Verify $z'^{(l)}$ by using the correct public key $\mathcal{P}$ as $h'^{(l)} := \mathcal{P}(z'^{(l)})$.
4: Set $\delta^{(l)} := h'^{(l)} - h^{(l)}$.

We denote with $l \in \mathbb{N}$ the iteration of the attack, i.e., in iteration l the l^{th} fault is induced and the l^{th} message is signed. Thus, $\delta^{(l)}$ is the difference between

[4] To clarify Step 2. of [14, page 9]: It is essential to cause a new fault on the central map for each message (i.e., for each iteration over Steps 1 - 4) and not use the same faulty map for all messages. Using the same faulty map for more than one message will not reveal new information about T, as for two messages $h^{(l_1)}$ and $h^{(l_2)}$ - signed with the same faulty central map - $\delta^{(l_1)}$ and $\delta^{(l_2)}$ will be multiples component-wise, since the attack would both times target the same column of T.

the l^{th} message and the message obtained from signing this message with the faulty central map $\mathcal{F}'$ and then applying the correct public key $\mathcal{P}$ to it. Hence, $\delta^{(l)}$ contains information about the difference between the correct and the faulty central map.

First, we show that during the whole attack it suffices to consider T, the linear part of $\mathcal{T}$, cf. Section 2.1. This is due to the fact that the constant part t cancels out, see Equation 2. For $z'^{(l)}$, the faulty signature in iteration l, correct public key $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{S}$, and faulty public key $\mathcal{P}' = \mathcal{T} \circ \mathcal{F}' \circ \mathcal{S}$, we have

$$\begin{aligned}
\delta^{(l)} &= h^{(l)} - h'^{(l)} = \mathcal{P}'(z'^{(l)}) - \mathcal{P}(z'^{(l)}) = (\mathcal{T} \circ \mathcal{F}' \circ \mathcal{S})(z'^{(l)}) - (\mathcal{T} \circ \mathcal{F} \circ \mathcal{S})(z'^{(l)}) \\
&= (\mathcal{T} \circ \mathcal{F}' \circ \mathcal{S}(z'^{(l)})) - (\mathcal{T} \circ \mathcal{F} \circ \mathcal{S}(z'^{(l)})) \\
&= [T(\mathcal{F}'((S(z'^{(l)}) + s))) + t] - [T(\mathcal{F}((S(z'^{(l)}) + s))) + t] \\
&= T(\mathcal{F}' - \mathcal{F})((S(z'^{(l)})) + s) \\
&= (T \circ (\mathcal{F}' - \mathcal{F}) \circ S)(z'^{(l)}).
\end{aligned} \tag{2}$$

Note that in the last three rows of Equation 2, we do not use $\mathcal{T}$ and $\mathcal{S}$, but T and S. Now we show how T is transformed: We assume that in the first iteration ($l = 1$), a coefficient $\xi^{(k_1)}$ in $\mathcal{F}$ is changed to $\xi^{(k_1)'}$. In the resulting difference between the correct and the faulty central map, there will be only one nonzero entry, exactly at position k_1: $(\mathcal{F}' - \mathcal{F})(x) = (0, \ldots, 0, (\xi^{(k_1)'} - \xi^{(k_1)}) x_i x_j, 0, \ldots, 0)^T$[5]. For the faulty signature $z'^{(1)}$ of the first message $h^{(1)}$, with Equation 2 we have:

$$\delta^{(1)} = (\delta_1^{(1)}, \ldots, \delta_m^{(1)}) = T \circ (\mathcal{F}' - \mathcal{F}) \circ S(z'^{(1)}) = T(0, \ldots, 0, c_1, 0, \ldots, 0)^T, \tag{3}$$

where c_1 at position k_1 is an unknown constant resulting from $S(z'^{(1)})$ plugged into $\mathcal{F}' - \mathcal{F}$. All other entries are zero, since the central map consists of m quadratic equations $f^{(1)}, \ldots, f^{(m)}$ and in the faulty central map only in the k_1^{th} equation one coefficient was changed by the fault. $\delta^{(1)}$ has length m and as we can see from Equation 3, it coincides with a constant multiple of the k_1^{th} column vector of the $m \times m$ matrix T. Hence, T can be written as $T = \begin{pmatrix} * \ldots * \, \delta_1^{(1)}/c_1 \, * \ldots * \\ \vdots \qquad \vdots \qquad \vdots \\ * \ldots * \, \delta_m^{(1)}/c_1 \, * \ldots * \end{pmatrix}$, where $(\delta_1^{(1)}/c_1, \ldots, \delta_m^{(1)}/c_1)^T$ is its k_1^{th} column. The idea is now to stepwise transform T into a triangular matrix. To do so, in each iteration l a matrix $T^{(l)}$ is multiplied to T, which by construction annihilates all entries in the k_i^{th} column except for the l^{th}. For the construction of this matrix, we define the vector

$$\delta(l) := (-\delta_{l+1}^{(l)}/\delta_l^{(l)}, \ldots, -\delta_m^{(l)}/\delta_l^{(l)})^T, \tag{4}$$

which has length $m - l$ in each step. Each matrix $T^{(l)}$ consists of four blocks, the sizes and structure of which change in each step depending on the value

[5] Note that this does not imply that one of the quadratic coefficients is changed. This representation only serves as an illustration.

l of the iteration. The upper left block is the $l \times l$ identity matrix, the upper right block consists of zeroes of dimension $l \times (m-l)$, the lower right block contains the $(m-l) \times (m-l)$ identity matrix and the lower left block, which has the size $(m-l) \times l$, includes the vector defined in Equation 4 in column l and a number of $l-1$ zero vectors of length $m-l$ in columns 1 to $l-1$, i.e.,

$$T^{(l)} = \left(\begin{array}{cc|c} & & [0]_{l\times(m-l)} \\ & & \\ \hline & \delta(l)_1 & \\ [0]_{(m-l)\times(l-1)} & \vdots & [I]_{(m-l)\times(m-l)} \\ & \delta(l)_{m-l} & \end{array}\right).$$ Hence, for $l = 1$ the matrix is

$$T^{(1)} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ -\delta_2^{(1)}/\delta_1^{(1)} & & & \\ \vdots & & [I_{m-1}] & \\ -\delta_m^{(1)}/\delta_1^{(1)} & & & \end{pmatrix} \text{ and } T^{(1)}T = \begin{pmatrix} * \cdots * & \delta_1^{(1)}/c_1 & * \cdots * \\ * \cdots * & 0 & * \cdots * \\ \vdots & \vdots & \vdots \\ * \cdots * & 0 & * \cdots * \end{pmatrix},$$

where $(\delta_1^{(1)}/c_1, 0, \cdots, 0)^T$ is the k_1^{th} column.
This calculation is performed at least $m-1$ times [6], until in the last step we have

$$T^{(m-1)} = \begin{pmatrix} & & 0 \\ & [I_{m-1}] & \vdots \\ & & 0 \\ 1 & 0 \cdots -\frac{\delta_m^{(m-1)}}{\delta_{m-1}^{(m-1)}} & 1 \end{pmatrix} \text{ and } T^{(m-1)}T = \begin{pmatrix} * \cdots * & \delta_1^{(m-1)}/c_{m-1} & * \cdots * \\ \vdots & \vdots & \vdots \\ * \cdots * & \delta_{m-1}^{(m-1)}/c_{m-1} & * \cdots * \\ * \cdots * & 0 & * \cdots * \end{pmatrix},$$

where $(\delta_1^{(m-1)}/c_{m-1}, \ldots, \delta_{m-1}^{(m-1)}/c_{m-1}, 0)^T$ is the ${k_{m-1}}^{th}$ column. If we put together T from the $m-1$ transformed matrices $T^{(1)}T, \ldots, T^{(m-1)}T$, we obtain a permutation of a triangular matrix with at most $\frac{m(m+1)}{2}$ nonzero entries. All other entries are expressed as quotients of some entry of δ^l and constants c_l.

By using the MinRank attack [16] we can now recover $\mathcal{T}$ by using the rank of the central equations $f^{(k)}$. The MinRank attack uses the fact that the rank of $\mathcal{F}^{(k)}$ is invariant under $\mathcal{S}$ (the transformation of variables), but changed by $\mathcal{T}$ (the transformation of equations). Since the entries of T have been reduced by this attack, the complexity of the MinRank attack is reduced as well [14].

3.3 UOV Schemes are Immune to this Attack

The authors of [14] state that the attack can be applied to UOV. However, in UOV the second affine map $\mathcal{T}$ can be omitted since using it does not increase the overall security of the scheme while increasing the key sizes and complexity [6].

[6] In Table 1 of [14] the authors state that the number of faults for STS type schemes - they erroneously consider UOV and Rainbow to be STS schemes - is exactly $n-1$. This is incorrect in two different ways: 1) According to the dimension of T, the number of faults does not depend on n, but on m. 2) The number of faults is not exactly $m-1$, but at least $m-1$. In Section 3.5, we describe a special case where more faults need to be injected .

This leaves us with a UOV public key of $\mathcal{F} \circ \mathcal{S}$. Thus, applying the proposed attack on a UOV scheme does not work, as the goal was to restore parts of the affine map $\mathcal{T}$. Interestingly, because of the different roles of the dimensions n and m and since $\mathcal{S}$ is computed before the central map in the public key, the attack can not be transferred to $\mathcal{S}$.

3.4 Applying the Attack to Rainbow Schemes

In this section we adapt the attack on the central map to a Rainbow scheme with parameters v_i for the vinegar variables and o_i for the oil variables with $i = 1, \ldots, u$, a central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ with $m = n - v_1$, and two affine maps $\mathcal{T} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^n$, see Appendix A.3.

First, we consider the case that the attacker does not know which map is affected by the fault and is only able to randomize values. Therefore, we compute the success probability for hitting a coefficient in the central map $\mathcal{F}$.

Success Probability The attack on the central map is only successful if actually an element in $\mathcal{F}$ is changed by the fault. However, we assume that the attacker can only randomly alter elements of either $\mathcal{S}$, $\mathcal{F}$, or $\mathcal{T}$ without knowing anything about the changed values. In order to estimate the success probability for hitting an element of the central map $\mathcal{F}$ we need to determine the number of all entries of the three matrices representing the maps $\mathcal{S}$, $\mathcal{F}$, and $\mathcal{T}$. We revise and detail the information hereof given in [14].

The affine map $\mathcal{T} : \mathbb{F}^m \to \mathbb{F}^m$ consists of a quadratic $m \times m$ matrix and a linear vector of length m. This gives a total of $m \cdot m + m = m(m+1)$ elements. Analogously, the affine map $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^n$ has a total of $n(n+1)$ elements. The central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^{n-v_1}$ contains m equations each theoretically in n variables. All variables that are not assigned in an equation, e.g., all terms of the form oil-oil, have the coefficient 0. The number of assigned variables depends on the layer. First, we provide the formula for the number of nonzero variables summed up over all layers: $\sum_{i=1}^{u} \frac{o_i(v_i+1)(v_i+2)}{2} - (n - v_i)$. This formula describes the actual combination of n variables quadratically, linearly, and constantly, considering that there are no oil-oil variables and the number of vinegar-vinegar variables depends on the layer.

In the general case, however, we assume that the attacker can change any of the coefficients stored in a coefficient matrix as depicted in [9, Figure 3], e.g., he he could "create" an oil-oil variable that does not exist (i.e., is zero) by changing the corresponding coefficient from 0 to another value. In the general case, for a single equation we have $n(n+1)/2$ quadratic terms, n linear terms, and one constant term. For m equations this sums up to a total of at most $m\frac{n(n+1)+2n+2}{2} = m\frac{(n+1)(n+2)}{2}$. Hence, we obtain the success probability[7]

$$p = \frac{m(n+1)(n+2)}{m(n+1)(n+2) + 2m(m+1) + 2n(n+1)}. \tag{5}$$

[7] The same formula holds for UOV schemes.

Since the parameter q does not appear in this formula, the success probability for this attack does not depend on the field $\mathbb{F}^8$. It rather depends on the ratio between the number of equations m and the number of variables n.

To learn the concrete success probability of the attack against the Rainbow scheme, we computed examples for different reasonable parameters. In [2], lower bounds for n, depending on the value of m, are given for finite fields with $q \in \{16, 31, 256\}$. For these fields, we selected four values for (m, n) from the literature [11, 20][9] and computed the success probability for the attack against schemes instantiated with theses parameters. The results are given in Table 1.

Rainbow parameters	success probability
$\mathbb{F}_{16}, m = 42, n = 61$	$p \sim 0.936$
$\mathbb{F}_{31}, m = 35, n = 52$	$p \sim 0.926$
$\mathbb{F}_{256}, m = 28, n = 48$	$p \sim 0.916$
$\mathbb{F}_{256}, m = 33, n = 27$	$p \sim 0.895$

Table 1: Success probability of hitting the central map in Rainbow schemes.

We conclude with the result that for Rainbow schemes in common fields and with up-to-date parameter choices, the success probability for hitting a coefficient in the central map $\mathcal{F}$ is more than 90%.

Assuming a Stronger Attacker On the one hand, a stronger attacker can target $\mathcal{F}$ (instead of only $\mathcal{P}$) or even specific coefficients in $\mathcal{F}$ directly. This allows him to perform the attack in a more structured way and to avoid unwanted scenarios. On the other hand, a stronger attacker can not only randomize values, but zero them or even set them to a chosen value. In case an attacker is more powerful in both ways, he can directly find values of $\mathcal{F}$: He chooses a random message and successively assigns all values from the underlying field to a certain entry of $\mathcal{F}$ before signing the message with that modified $\mathcal{F}$. As soon as a $\delta^{(l)}$ consists of only zeroes, the right entry of $\mathcal{F}$ is found.

3.5 Special Cases

During anaylzing how the attack can be applied to a Rainbow scheme, we detected some special cases that can occur and which are not covered by the descriptions in [14].

Specific Vinegar Variable Assigned 0 In each signature generation process there are a number of values randomly assigned to the vinegar variables over the field $\mathbb{F}$.

[8] Actually the parameters q and n are indirectly connected, since in fields with small q the parameter n has to be chosen larger in order to ensure security.

[9] The first three tuples of parameters are taken from [20] for the year 2018, the last one is the original suggestion from [11].

Let us assume v_i to be the number of vinegar variables. If the coefficient changed by the fault attack belongs to a variable that contains a vinegar-monomial (i.e., vinegar-vinegar or vinegar-oil) and furthermore exactly this vinegar variable takes the value 0 during the step in the inversion of the central map where random values are assigned to the vinegar variables, then this term with the faulty coefficient drops out during signature generation. As a consequence, there is no difference in a signature generated with the correct central map and one generated with the faulty central map, resulting in $\delta^{(k)} = 0$ in all entries. When the attacker computes $\delta^{(k)} = 0$, he realizes that this case occurred[10] but gains no information in the sense of the attack.

We computed the probability for this special case to occur for an example Rainbow scheme. In the original paper of Ding and Schmidt [11], a set of parameters for practical implementation is proposed. For these parameters, we derived a probability of approximately 1.1%, as explained in Appendix B.

l^{th} Entry of $\delta^{(l)}$ Equals 0 The second special case concerns that in each iteration l of the attack $\delta^{(l)} = h'^{(l)} - h^{(l)}$ is computed, with $\delta^{(l)} = (\delta_1^{(l)}, \ldots, \delta_m^{(l)})$. These entries are a constant multiple of the k_l^{th} column of the matrix T. This implies that if the k_l^{th} entry of a column in T equals 0, then for the l^{th} entry of $\delta^{(l)}$ it holds $\delta_l^{(l)} = 0$. In this case it is not possible to construct the vector in Equation 4 in order to perform the transformation since all other entries of $\delta^{(l)}$ would have to be divided by $\delta_l^{(l)} = 0$. An attacker would detect this occurrence by computing $\delta^{(l)}$, but could discard the values and start over.

The probability for this special case to occur depends on m, the number of multivariate quadratic polynomials, and on the size of the finite field. For each column separately, the probability is $\frac{1}{|\mathbb{F}|}$, since the values in T were assigned randomly from $\mathbb{F}$. Performing this step at least $m-1$ times, the probability p_2 for this special case to occur can be computed via the complementary event: $p_2 \geq 1 - \left(\frac{|\mathbb{F}|-1}{|\mathbb{F}|}\right)^{m-1}$. For the example schemes from Table 1, this yields $p_2 \geq 0.928$ ($q = 16$, $m = 42$), $p_2 \geq 0.672$ ($q = 31, m = 35$), and $p_2 \geq 0.100$ ($q = 256, m = 28$). With increasing field size, the probability decreases drastically.

Coefficients in Same Equation Targeted More Than Once The third special case concerns redundant faults: It can happen that an attacker injects faults that affect an equation that had already been altered with a previous fault, i.e., the same column of T is affected several times. The attacker would detect this situation if a newly computed $\delta^{(k)}$ is linearly dependent to any of the already computed ones. Since the goal is to transform T into a triangular matrix where the k_l^{th} column vector contains information about the k_l^{th} column of T, it is necessary to target each equation (at least) once. Hence, an attacker would abort this step and try to target another equation which is yet untouched.

[10] The same situation would occur if an entire column of $\mathcal{T}$ was equal zero. However, this cannot happen since the maps are expected to have full rank.

4 Fault Attack on the Random Values

In this section we show how the attack on the random values can be applied to the SingleField schemes UOV and Rainbow. First, we introduce the attacker model in Section 4.1. Then, we explain how the attack can be applied to SingleField signature schemes. The explanation of the attack method is similar to the description in [14, Section 3.3.2] with a slightly different notation and more details. In Section 4.3, we discuss a special case of the attack that has not been covered in [14], and from this deduce the success probability of the attack.

4.1 Attacker Model

In each signature generation the vinegar variables are instantiated with random values. In this attack, which targets the signature generation process, we assume that the attacker fixes some (or all) of these random values with a single permanent fault. He does not know how many variables he fixed, and he does not know the value of these variables. Afterwards, he receives several message-signature pairs where each signature has been computed with the fixed variables and, in case he did not fix all of the variables, additional random ones. The more variables he fixes, the less message-signature pairs he needs. By analyzing these pairs, the attacker gains partial information of $\mathcal{S}$.

4.2 Detailed Description of the Attack for SingleField Schemes

We denote the random values with $r_1, \ldots, r_{u_1} \in \mathbb{F}, u_1 \in \mathbb{N}$ and assume that the attacker fixes the first u_2 variables $r_1, \ldots, r_{u_2}$ for $u_2 \leq u_1$.

Algorithm 2 Attack on the random values

1: Cause a fault that fixes $r_1, \ldots, r_{u_2}$ and suppose that $\bar{r}_1, \ldots, \bar{r}_{u_2} \in \mathbb{F}$ are exactly these unknown fixed values.
2: Generate signatures $z^{(1)}, \ldots, z^{(n-u_2+1)}$ for randomly chosen messages $h^{(1)}, \ldots, h^{(n-u_2+1)}$ with $r = (\bar{r}_1, \ldots, \bar{r}_{u_2}, r_{u_2+1}, \ldots, r_n)$.
3: Recover parts of $\mathcal{S}$ by using the pairs $(z^{(k)}, h^{(k)})$.

Consider a UOV scheme over a finite field $\mathbb{F}$ with v vinegar and o oil variables satisfying $v > o$ or a Rainbow scheme with v_i vinegar and o_i oil variables per layer $i = 1, \ldots, u$. Since the attack works analogously for both schemes, we simply write v. Let $h^{(1)}, \ldots, h^{(n-u_2+1)} \in \mathbb{F}^m$ be the messages and $z^{(1)}, \ldots, z^{(n-u_2+1)} \in \mathbb{F}^n$ the corresponding signatures with $u_2 \leq v$ variables that have been fixed by the attacker. Let $x^{(k)} = (x_1^{(k)}, \ldots, x_v^{(k)}) \in \mathbb{F}^v$ be the vinegar variables in step k. W.l.o.g. we assume that the first u_2 variables $(x_1, \ldots, x_{u_2})$ are fixed to the values $(\bar{x}_1 \ldots, \bar{x}_{u_2})$, yielding the vector $x^{(k)} = (\bar{x}_1 \ldots, \bar{x}_{u_2}, x_{u_2+1}^{(k)}, \ldots, x_v^{(k)})^T$ for each step k. We write $x^{(k)} = (\bar{x}, r^{(k)})^T$, where $\bar{x}$ denotes the fixed part and $r^{(k)} \in \mathbb{F}^{v-u_2}$ denotes the random values that differ in each step.

Below we will show that a total of $n - u_2 + 1$ message-signature pairs are needed to perform the attack.

Reducing the Number of Nonzero Elements in a Specific Representation of S Signatures in UOV and Rainbow are computed by $z = \mathcal{S}^{-1}(\mathcal{F}^{-1}(y))$ and $z = \mathcal{S}^{-1}(\mathcal{F}^{-1}(\mathcal{T}^{-1}(y)))$, respectively. In both cases we can write

$$z = \mathcal{S}^{-1}\begin{pmatrix} x \\ w \end{pmatrix}$$

for some $x \in \mathbb{F}^v$ and $w \in \mathbb{F}^{n-v}$. With the above notation we can rewrite

$$Sz^{(k)} + s = \begin{pmatrix} \bar{x} \\ r^{(k)} \\ w^{(k)} \end{pmatrix}. \tag{6}$$

We want to see how the fixed values $\bar{x}$ can be used to express S, so we split up S, $z^{(k)}$, and s into

$$S = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \text{ and } z^{(k)} = \begin{pmatrix} z^{(k,1)} \\ z^{(k,2)} \end{pmatrix} \text{ and } s = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix},$$

where $z^{(k,1)}, s_1 \in \mathbb{F}^{u_2}$ and $z^{(k,2)}, s_2 \in \mathbb{F}^{n-u_2}$ and $A \in \mathbb{F}^{u_2 \times u_2}$, $B \in \mathbb{F}^{u_2 \times (n-u_2)}$, $C \in \mathbb{F}^{(n-u_2)\times u_2}$, and $D \in \mathbb{F}^{(n-u_2)\times(n-u_2)}$. We now use Equation 6 to write

$$Sz^{(k)} + s = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} z^{(k,1)} \\ z^{(k,2)} \end{pmatrix} + \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} Az^{(k,1)} + Bz^{(k,2)} + s_1 \\ Cz^{(k,1)} + Dz^{(k,2)} + s_2 \end{pmatrix} = \begin{pmatrix} \bar{x} \\ r^{(k)} \\ w^{(k)} \end{pmatrix}.$$

From the dimensions of A, B, C, and D we deduce $Az^{(k,1)} + Bz^{(k,2)} + s_1 = \bar{x}$. As s_1 and $\bar{x}$ are fixed from the beginning, we write $Az^{(k,1)} + Bz^{(k,2)} = \bar{x} - s_1$ and with setting $\bar{z}^{(k,1)} := z^{(k,1)} - z^{(1,1)}$ and $\bar{z}^{(k,2)} := z^{(k,2)} - z^{(1,2)}$ for $2 \leq k \leq n - u + 1$, we obtain $A\bar{z}^{(k,1)} + B\bar{z}^{(k,2)} = A(z^{(k,1)} - z^{(1,1)}) + B(z^{(k,2)} - z^{(1,2)}) = Ax^{(k,1)} - Bz^{(k,2)} - (Az^{(1,1)} + Bz^{(1,2)}) = \bar{x} - s_1 - (\bar{x} - s_1) = 0$. Based on this we are able to express $A^{-1}B$ with the aid of the signatures $z^{(k)}$ by using $\bar{z}^{(k,1)}$, $k \in \{2, \ldots, n - u_2 + 1\}$, as column $k-1$ of the $u_2 \times (n - u_2)$-matrix Z_1 and accordingly $\bar{z}^{(k,2)}$ as column $k-1$ of the $(n-u_2) \times (n-u_2)$-matrix Z_2. It follows

$$AZ_1 + BZ_2 = 0 \Leftrightarrow AZ_1 = -BZ_2 \Leftrightarrow Z_1 = -A^{-1}BZ_2 \Leftrightarrow -Z_1{Z_2}^{-1} = A^{-1}B \tag{7}$$

if A and Z_2 are invertible. The facilitated representation of S is then given by $\begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \begin{pmatrix} I_{u_2} & -A^{-1}B \\ 0 & I_{n-u_2} \end{pmatrix} = \begin{pmatrix} A & 0 \\ C & -CA^{-1}B + D \end{pmatrix}$. Hence, the attack on the random values can be used to reduce the number of nonzero elements in the facilitated representation of S. Subsequently, the MinRank attack [16] can be used to compute S [12].

4.3 Special Case and Success Probability of the Attack

The attack does not work if A is a singular matrix, as can be seen in Equation 7. To discuss the probability of this special case, we determine the probability that

an $(n \times n)$-matrix with random entries from $\mathbb{F}$ is invertible, i.e., not singular. Following [7], we estimate this probability under the assumption that the entries are uniformly distributed in $\mathbb{F}$ as

$$\prod_{i=0}^{n-1} \frac{(q^n - q^i)}{q^{n^2}} = \prod_{i=1}^{n} (1 - \frac{1}{q^i}).$$

In the attack, matrix A has dimension $u_2 \times u_2$, where u_2 is the number of random variables that the attacker fixed. For common parameters for UOV and Rainbow schemes, we get high success probabilities that A is invertible, see Table 2. With u_2 increasing, the probability decreases only slightly.

finite field	number of fixed vinegar variables	success probability
$\mathbb{F}_{16}$	$u_2 \in \{1, \ldots, 16\}$	$p \geq 0.933$
$\mathbb{F}_{31}$	$u_2 \in \{1, \ldots, 31\}$	$p \geq 0.966$
$\mathbb{F}_{256}$	$u_2 \in \{1, \ldots, 256\}$	$p \geq 0.996$

Table 2: Success probability that the matrix $A \in \mathbb{F}^{u_2 \times u_2}$ is invertible, depending on different sizes of the finite field $\mathbb{F}$, as suggested for UOV and Rainbow [20], and different numbers u_2 of fixed vinegar variables.

On the other hand, the number of fixed values u_2 affects how many messages need to be signed, cf. Section 4.2. This is related to the complexity of the MinRank attack (which is used to learn S completely) which initially is $\mathcal{O}(q^{v-o-1}o^4) = \mathcal{O}(q^{n-2o-1}o^4)$ [5] with $n = v + o$ For each fixed vinegar variable the complexity is reduced by the factor q, i.e., in total by q^{u_2}. Hence, if an attacker fixes a number u_2 of vinegar variables, the complexity decreases to $\mathcal{O}(q^{(v-u_2)-o-1}o^4) = \mathcal{O}(q^{n-2o-u_2-1}o^4)$.

Consequently, an attacker should fix as many vinegar variables as possible.

5 Countermeasures

Derived from the fault attacks explained in the previous sections, we present algorithmic countermeasures to protect multivariate SingleField signature schemes against these attacks.

5.1 Securing the Central Map

Check for a Faulty Central Map An approach that has already been proposed in [14] is to test the central map for modifications before starting signature generation. The idea is to store a checksum $c_{\mathcal{F}}$ of the coefficients in $\mathcal{F}$ and compare it at the beginning of each signature generation with a checksum $c_{\mathcal{F}'}$

of the coefficients of the central map used during that signature generation. In case the checksums differ, the message is not signed. This countermeasure can be applied to all SingleField schemes. However, the checking procedure has to be carefully implemented, i.e., protected, so that it cannot be skipped by an experienced attacker [3].

Increase the Chances for Vinegar Variables to be 0 As shown in Section 3.5, a situation can occur where the faulty coefficient in the central map coincides with the choice of a vinegar variable to be 0 during the signature generation process. In this case the whole expression with the faulty coefficient and the vinegar variable evaluates to 0. We learned that each time this happens, the attacker has to start over again since this step does not yield new information. The idea of this countermeasure is to increase the probability of the vinegar variables to be assigned 0 in order to increase the overall probability for it to coincide with the exact faulty coefficient. (This of course requires the faulty coefficient to be of vinegar-type.) The vinegar variables are assigned with random values from the underlying finite field $\mathbb{F}$. Hence, signature schemes that use smaller finite fields are better protected against the attack on the central map.

Increase the Number of 0-entries in T As discussed in Section 3.5, it can happen that in the l^{th} iteration the l^{th} entry of $\delta^{(l)}$ equals 0. Then the attacker cannot proceed with the attack, since in order to reduce the elements of T it is necessary to divide all other entries of $\delta^{(l)}$ by $\delta_l^{(l)}$. To make the attack less likely to work, we can thus increase the number of 0-entries in T, so that it gets more probable to have such a 0-entry at the according position. However, there are several problems involved: Too many 0-entries result in sparseness of the matrix and while a sparse matrix might impede this attack, it simultaneously facilitates rank attacks. Also, when the attacker learns which entries are 0 he might use this knowledge to adjust the attack accordingly. We leave for future work to analyze if indeed it is reasonable to increase the number of 0-entries in T.

Change the Ratio of m and n In Section 3.4 we showed that the success probability for changing an entry in $\mathcal{F}$ is around 90% for Rainbow schemes. This high probability comes from the fact that the size of $\mathcal{F}$ is relatively large in comparison to $\mathcal{T}$ and $\mathcal{S}$. This depends on the ratio of n and m. So it seems to be a reasonable idea to make the attack less successful by changing the ratio of the variables m and n and thus increase the probability that an attacker targets parameters of $\mathcal{T}$ or $\mathcal{S}$ instead of $\mathcal{F}$. This can be achieved by minimizing Equation 5. However, if an attacker is able to distinguish the faulty place (c.f. Scenario 2 in Section 3.4), he realizes if the fault injection was successful and can repeat the attack in case it was not. Again, we leave for future work to determine how this countermeasure impairs the security of the scheme. This countermeasure is also applicable for other schemes of SingleField type.

5.2 Securing the Random Values

Saving the Values The first idea to prevent the attack on the random values has already been roughed out in [14]. This countermeasure consists in saving the randomly chosen values for each step and compare them with the variables of every current signature. If a certain threshold of coincidences between old and new values occurs, the signature generation has to be aborted. The countermeasure can be applied to all SingleField schemes which use random values.

This threshold has to be chosen carefully, since, as we show in Appendix C, also without fault injection coincidences are frequent. The choice of this threshold depends, among others, on the underlying field: the smaller the field, the more likely a coincidence in the random variables occurs. Considering the specifics of the attack, it might moreover be reasonable to count coincidences column-wise and abort further signature generations once the threshold is reached in one of the columns.

Matrix A not Invertible In Equation 7 we showed that the matrix A is required to be invertible, otherwise the transformation of S to reduce the number of nonzero elements does not work. A is the upper left part of the matrix S with dimension $u_2 \times u_2$, with $0 \leq u_2 \leq v$, where u_2 is the number of fixed variables. A powerful attacker would try to fix as many variables as possible. Since we do not know the value of u_2, but u_2 is bounded above by v, this countermeasure consists in filling the upper v entries of the first column of S with zeroes and thereby force A to be singular without necessarily making S singular (as $v < n$). Although this countermeasure completely prevents the attack against the random values, we leave for future work to analyze any security implications this might entail.

6 Conclusion

With this paper, we complement the research on the physical attack security of multivariate signature schemes. We presented to fault attacks on SingleField schemes with an emphasis on UOV and Rainbow. We showed that the success probability of both attacks is rather high. Nevertheless, since both attacks do not lead to complete key recovery, we conclude that multivariate signature schemes inherently offer a good protection against fault attacks.

Acknowledgments

This work has been co-funded by the DFG as part of project P1 within the CRC 1119 CROSSING. We thank Mohamed Saied Emam Mohamed for his contribution to a preliminary version of this work and Albrecht Petzold for his diligent proofreading of this paper.

References

1. Round 2 submissions - post-quantum cryptography — CSRC (2019), https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions, visited on February 14, 2019
2. Albrecht, Bulygin, S., Buchmann, J.A.: Selecting parameters for the rainbow signature scheme - extended version -. IACR Cryptology ePrint Archive **2010**, 437 (2010)
3. Blömer, J., da Silva, R.G., Günther, P., Krämer, J., Seifert, J.P.: A practical second-order fault attack against a real-world pairing implementation. 2014 Workshop on Fault Diagnosis and Tolerance in Cryptography pp. 123–136 (2014)
4. Bogdanov, A., Eisenbarth, T., Rupp, A., Wolf, C.: Time-area optimized public-key engines: MQ-cryptosystems as replacement for elliptic curves? In: Cryptographic Hardware and Embedded Systems – CHES 2008. pp. 45–61. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
5. Braeken, A., Wolf, C., Preneel, B.: A study of the security of unbalanced oil and vinegar signature schemes. In: CT-RSA. Lecture Notes in Computer Science, vol. 3376, pp. 29–43. Springer (2005)
6. Bulygin, S., Petzoldt, A., Buchmann, J.A.: Towards provable security of the unbalanced oil and vinegar signature scheme under direct attacks. In: Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6498, pp. 17–32. Springer (2010)
7. Charlap, L.S., Rees, H.D., Robbins, D.P.: The asymptotic probability that a random biased matrix is invertible. Discrete Mathematics **82**(2), 153–163 (1990)
8. Chen, A.I.T., Chen, M.S., Chen, T.R., Cheng, C.M., Ding, J., Kuo, E.L.H., Lee, F.Y.S., Yang, B.Y.: SSE Implementation of Multivariate PKCs on Modern x86 CPUs. In: Cryptographic Hardware and Embedded Systems - CHES 2009. pp. 33–48. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
9. Czypek, P., Heyse, S., Thomae, E.: Efficient Implementations of MQPKS on Constrained Devices. In: Cryptographic Hardware and Embedded Systems – CHES 2012. pp. 374–389. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
10. Ding, J., Chen, M., Petzoldt, A., Schmidt, D., Yang, B.: Rainbow - algorithm specification and documentation (November 2017), https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions
11. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3531, pp. 164–175 (2005)
12. Faugère, J., Levy-dit-Vehel, F., Perret, L.: Cryptanalysis of minrank. In: Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5157, pp. 280–296. Springer (2008)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA (1990)
14. Hashimoto, Y., Takagi, T., Sakurai, K.: General fault attacks on multivariate public key cryptosystems. In: Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7071, pp. 1–18. Springer (2011)

15. Hashimoto, Y., Takagi, T., Sakurai, K.: General fault attacks on multivariate public key cryptosystems. IEICE Transactions **96-A**(1), 196–205 (2013)
16. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. Lecture Notes in Computer Science, vol. 1592, pp. 206–222. Springer (1999)
17. Kipnis, A., Shamir, A.: Cryptanalysis of the oil & vinegar signature scheme. In: Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings. Lecture Notes in Computer Science, vol. 1462, pp. 257–266. Springer (1998)
18. Okeya, K., Takagi, T., Vuillaume, C.: On the importance of protecting Δ in SFLASH against side channel attacks. International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. **2**, 560–568 Vol.2 (2004)
19. Park, A., Shim, K.A., Koo, N., Han, D.G.: Side-channel attacks on post-quantum signature schemes based on multivariate quadratic equations. IACR Transactions on Cryptographic Hardware and Embedded Systems **2018**(3), 500–523 (Aug 2018)
20. Petzoldt, A.: Selecting and reducing key sizes for multivariate cryptography. Ph.D. thesis, Darmstadt University of Technology, Germany (2013)
21. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
22. Steinwandt, R., Geiselmann, W., Beth, T.: A theoretical dpa-based cryptanalysis of the NESSIE candidates FLASH and SFLASH. In: ISC. Lecture Notes in Computer Science, vol. 2200, pp. 280–293. Springer (2001)
23. Tang, S., Yi, H., Ding, J., Chen, H., Chen, G.: High-Speed Hardware Implementation of Rainbow Signature on FPGAs. In: Post-Quantum Cryptography. pp. 228–243. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
24. Yi, H., Li, W.: On the Importance of Checking Multivariate Public Key Cryptography for Side-Channel Attacks: The Case of enTTS Scheme. The Computer Journal **60**(8), 1197–1209 (2017)
25. Yi, H., Nie, Z.: High-speed hardware architecture for implementations of multivariate signature generations on FPGAs. EURASIP Journal on Wireless Communications and Networking **2018**(1), 93 (May 2018)

A The Signature Schemes UOV and Rainbow

A.1 Signature Generation and Verification of Multivariate Schemes

The standard signature generation and verification process of a multivariate signature scheme works as shown in Figure 1.

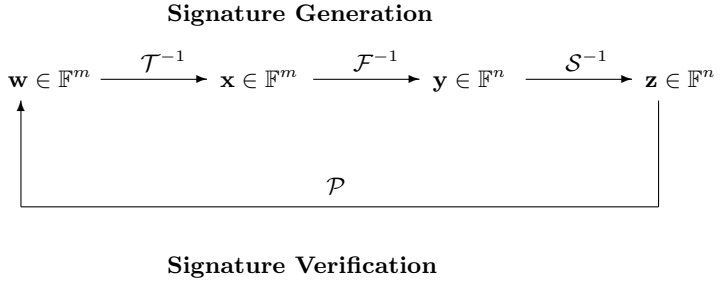

Fig. 1: General workflow of multivariate signature schemes.

A.2 Unbalanced Oil and Vinegar Signature Scheme

The Unbalanced Oil and Vinegar signature scheme (UOV) is a modified version of the Oil and Vinegar scheme. It was designed by Kipnis and Patarin and presented at EUROCRYPT'99 [16] after the original scheme was broken by Kipnis and Shamir in 1998 [17] via linear algebra attacks.

Unlike in the Oil and Vinegar scheme, where the number of vinegar and oil variables are equal, the advantage of UOV consists in choosing the number of vinegar variables to be greater than the number of oil variables in order to guarantee better security against known attacks. The formal notation, the choice of variables, and the structure of the scheme is described in the following.

Notation All computations are performed in a finite field $\mathbb{F}$ with q elements. Let $o := m \in \mathbb{N}$ be the number of oil variables and $v \in \mathbb{N}$ the number of vinegar variables, hence $n = o + v$. The corresponding index sets for the variables be $V = \{1,...,v\}$ and $O = \{v+1,...,n\}$. $x_i (i \in V)$ are called vinegar variables and $x_j (j \in O)$ oil variables. The message (or its hash) to be signed is denoted by $h = (h_1,...,h_m) \in \mathbb{F}^m$ and the signature itself by $z = (z_1,...,z_n) \in \mathbb{F}^n$.

Central Map and Affine Maps The central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^o$ of the UOV-scheme consists of m quadratic polynomials $f^{(1)},...,f^{(m)} \in \mathbb{F}[x_1,...,x_n]$ of the

form

$$f^{(k)}(x) = \sum_{\substack{i,j \in V \\ i \leq j}} {\alpha_{ij}}^{(k)} x_i x_j + \sum_{\substack{i \in V \\ j \in O}} {\beta_{ij}}^{(k)} x_i x_j + \sum_{i \in V \cup O} {\gamma_i}^{(k)} x_i + \eta^{(k)}$$

where $k \in \{1,...,m\}$ and the $\alpha_{ij}^{(k)}$ are the coefficients of the quadratic vinegar-vinegar, the $\beta_{ij}^{(k)}$ of the quadratic oil-vinegar, the $\gamma_i^{(k)}$ of the linear oil and vinegar variables and $\eta^{(k)}$ is the constant part. All coefficients are chosen randomly from the underlying field $\mathbb{F}$ and stored in a matrix, see, e.g., [9, Figure 2].

In order to hide the structure of the central map $\mathcal{F}$, it is composed with an affine bijective map $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^n$, which can be written as $\mathcal{S}(x) = Sx + s$, where $S \in \mathbb{F}^{n \times n}$ is a linear transformation and $s \in \mathbb{F}^n$ is a vector.

Note that unlike in other multivariate signature schemes of SingleField type like Rainbow (cf. Section A.3), in UOV the second affine map $\mathcal{T} : \mathbb{F}^m \to \mathbb{F}^m$ can be omitted (or similarly treated like the identity map $\mathcal{T} = id$) since applying it to the polynomials would not change the structure of the central map $\mathcal{F}$ at all and thus would not increase the overall security.

Public Key and Private Key The public key of the UOV scheme is given by $\mathcal{P} = \mathcal{F} \circ \mathcal{S}$ with $\mathcal{P} : \mathbb{F}^n \xrightarrow{\mathcal{S}} \mathbb{F}^n \xrightarrow{\mathcal{F}} \mathbb{F}^m$, consisting of m public quadratic polynomials in n variables. The private key is the tuple $(\mathcal{F}, \mathcal{S})$. As both $\mathcal{F}$ and $\mathcal{S}$ can be inverted efficiently, knowledge of the private key allows for inversion of the public key and therefore signature generation.

Inversion of the Central Map In order to create a valid signature, inversion of the central map is required (compare Equation 8 below), which is done by performing the following steps:

1. Assign random values to the vinegar variables $x_1,...,x_v$.
2. Substitute them into the polynomials $f^{(1)},...,f^{(m)}$, resulting in a system of m linear equations in the oil variables $x_{v+1},...,x_n$.
3. Solve the system of linear equations, e.g., by using Gaussian elimination.
4. If the system does not have a solution, go back to Step 1 and try again with different random values.

Signature Generation and Verification To sign a document $h = (h_1, \ldots, h_m) \in \mathbb{F}^m$, solve the equation

$$\mathcal{F} \circ \mathcal{S} z = h$$

for $z \in \mathbb{F}^n$. First, find a pre-image of h under the central map $\mathcal{F}$ with the method described above to get

$$\mathcal{S} z = \mathcal{F}^{-1} h =: y \tag{8}$$

with $y \in \mathbb{F}^n$. Then invert $\mathcal{S}$ to obtain the signature

$$z = \mathcal{S}^{-1} y.$$

For signature verification it has to be checked whether $\mathcal{P}(z) = h$ holds. If this is the case, the signature is accepted, if not, rejected.

A.3 Rainbow

In 2005, Ding and Schmidt published a new signature scheme named Rainbow, which is a generalization of the Unbalanced Oil and Vinegar scheme [11]. The basic idea is to combine several layers of Oil and Vinegar in one scheme in order to improve the security and efficiency of the scheme. Compared to UOV, in Rainbow key and signature sizes can be reduced.

Notation Let $\mathbb{F}$ be a finite field with q elements. Let S be the set $\{1,...,n\}$ and $v_1,...,v_{u+1}$ integers with the property

$$0 < v_1 < v_2 < ... < v_{u+1} = n,$$

where u stands for the number of layers. Define the sets of integers $S_i = \{1,...,v_i\}$ for each $i = 1,...,u$. The number of elements in set S_i is v_i and by construction we have

$$S_1 \subset S_2 \subset ... \subset S_{u+1} = S.$$

We set $o_i := v_{i+1} - v_i$ and $O_i := S_{i+1} - S_i = \{v_i + 1,...,v_{i+1}\}$ for $i = 1,...,u$. Then we have $|O_i| = o_i$.

Central Map and Affine Maps The central map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^{n-v_1}$, which is an easily invertible quadratic map, consists of $m := n - v_1$ polynomials $(f^{(v_1+1)},...,f^{(n)})$, each of the form

$$f^{(k)}(x_1,...,x_n) = \sum_{\substack{i,j \in S_l \\ i \leq j}} {\alpha_{ij}}^{(k)} x_i x_j + \sum_{\substack{i \in O_l \\ j \in S_l}} {\beta_{ij}}^{(k)} x_i x_j + \sum_{i \in O_l \cup S_l} {\gamma_i}^{(k)} x_i + \eta^{(k)}$$

with $k = v_1 + 1,...,n$ and where l denotes the layer. For $i \in O_l$ we call x_i an l^{th}-layer oil variable and for $i \in S_l$ an l^{th}-layer vinegar variable. The central map of a Rainbow scheme consists of u different layers, the i^{th} layer of which consists of the polynomials $f^{(j)}$ for $j \in O_i$.

The name Rainbow refers to the fact that the number of variables increases with each layer and can be arranged like the layers of a rainbow:

$$\begin{array}{l} [x_1,...,x_{v_1}]\{x_{v_1+1},...,x_{v_2}\} \\ [x_1,...,x_{v_1},x_{v_1+1},...,x_{v_2}]\{x_{v_2+1},...,x_{v_3}\} \\ \quad \vdots \qquad\qquad\qquad\qquad\qquad \vdots \\ [x_1,...,...,...,...,...,...,...,...,...,...,...,...,x_{v_{u-1}}]\{x_{v_{u-1}+1},...,x_n\}. \end{array}$$

Each row represents a layer of the Rainbow scheme with the vinegar variables in squared and the oil variables in curly brackets.

In order to hide the structure of the central map, two invertible affine maps are composed to $\mathcal{F}$ from both sides:

$$\begin{aligned} \mathcal{S} &: \mathbb{F}^n \to \mathbb{F}^n \quad \text{with } \mathcal{S}(x) = Sx + s \text{ for } x \in \mathbb{F}^n \\ \mathcal{T} &: \mathbb{F}^m \to \mathbb{F}^m \quad \text{with } \mathcal{T}(y) = Ty + t \text{ for } y \in \mathbb{F}^m, \end{aligned} \tag{9}$$

where $T \in \mathbb{F}^{m \times m}$ and $S \in \mathbb{F}^{n \times n}$ are linear transformations and $t \in \mathbb{F}^m$ and $s \in \mathbb{F}^n$ are constant vectors.

Public Key and Private Key The public key is given by $\mathcal{P} = \mathcal{T} \circ \mathcal{F} \circ \mathcal{S}$ with $\mathcal{P} : \mathbb{F}^n \xrightarrow{\mathcal{S}} \mathbb{F}^n \xrightarrow{\mathcal{F}} \mathbb{F}^m \xrightarrow{\mathcal{T}} \mathbb{F}^m$. The field $\mathbb{F}$ and its additive and multiplicative structure are also publicly known. The private key consists of $(\mathcal{T}, \mathcal{F}, \mathcal{S})$.

Inversion of the Central Map In order to generate a signature, one needs to be able to invert $\mathcal{F}$. This can be done by the following steps, similar to the method for UOV, cf. Section A.2.

1. Assign values to the vinegar variables $x_1, \ldots, x_{v_1}$ at random and substitute them into the equations given by $f^{(v_1+1)}, \ldots, f^{(n)}$.
2. Solve the system of o_1 linear equations in the o_1 unknowns $x_{v_1+1}, \ldots, x_{v_2}$, e.g., via Gaussian elimination. This gives all the values x_i with $i \in S_2$.
3. Insert these values into the second layer of polynomials (i.e., $f^{(k)}$ with $k > v_2$) to obtain a system of o_2 linear equations in the o_2 unknowns $x_i, i \in O_2$. Solving the systems yields the x_i with $i \in S_3$.
4. Repeat this process until a solution for all variables is found. If in any step no solution for the systems of equations can be found, again random values for the variables $x_1, \ldots, x_{v_1}$ are chosen.

Signature Generation and Verification To sign a document $h = (h_1, \ldots, h_m) \in \mathbb{F}^m$, the equation

$$\mathcal{T} \circ \mathcal{F} \circ \mathcal{S}(z_1, \ldots, z_n) = h$$

needs to be solved for $z = (z_1, \ldots, z_n)$. To do this, first the inverse $\mathcal{T}^{-1}$ is applied

$$\mathcal{F} \circ \mathcal{S} z = \mathcal{T}^{-1} h =: x.$$

Next invert the central map $\mathcal{F}$ via the method described above to get

$$\mathcal{S} z = \mathcal{F}^{-1} x =: y.$$

Finally apply the inverse $\mathcal{S}^{-1}$ to obtain a signature z

$$z = \mathcal{S}^{-1} y.$$

To verify a signature one simply checks whether $\mathcal{P}(z) = h$ holds. In this case the signature is accepted, otherwise rejected.

B Probability for the Special Case of Section 3.5

We are interested in the following case: A fault is caused on a coefficient of the multivariate system. Coincidentally, during the signature generation process, a vinegar variable belonging to this coefficient is assigned 0.

The probability of a certain vinegar variable x_i to be assigned 0 is $\frac{1}{q}$, where $q = |\mathbb{F}|$. So the probability that at least one variable is chosen 0 is $1 - (1 - \frac{1}{q})^{v_l}$, where l is a layer in the Rainbow scheme. In the system of equations we have quadratic and linear terms with vinegar variables. The number of terms in the central map including a certain vinegar variable is $n + 1$ for one equation or $m(n + 1)$ for the whole system of equations. The total number of terms in the system consisting of $\mathcal{S}, \mathcal{T}$ and $\mathcal{F}$ thereby is given by $n(n + 1)$, $m(m + 1)$ and $\frac{m(n+1)(n+2)}{2}$, respectively.

The probability p for all u layers is then computed by

$$p = \left(\sum_{l=1}^{u} (1 - (1 - \frac{1}{q})^{v_l}) \right) \cdot \frac{m(n+1)}{m(m+1) + n(n+1) + \frac{m(n+1)(n+2)}{2}}$$

To get an idea of the concrete probability, we apply the considerations above to an example Rainbow scheme. Ding and Schmidt proposed in the original paper [11] a set of parameters for practical implementation. The finite field has $q = 2^8$ elements and $n = 33, S = \{1,2,...,33\}$. The number of layers is given by $u = 4$, the number of vinegar variables by $v_1 = 6, v_2 = 12, v_3 = 17, v_4 = 22, v_5 = 33$, and the number of oil variables by $o_1 = 6, o_2 = 5, o_3 = 5, o_4 = 11$, $m = n - v_1 = 27$. This yields:

$$p = \left(\sum_{l=1}^{4} (1 - (1 - \frac{1}{256})^{v_l}) \right) \cdot \frac{27 \cdot 34}{27 \cdot 28 + 33 \cdot 34 + \frac{27 \cdot 34 \cdot 35}{2}} \approx 0.011.$$

Hence, with the parameter choice given above, this special case approximately occurs in 1.1% of signature generations.

parameters	$p_1 \approx$	$i \geq 2 \rightarrow p_2 \geq$
UOV($\mathbb{F}_{16}, v = 128$)	0.99	0.99
UOV($\mathbb{F}_{31}, v = 104$)	0.97	0.99
UOV($\mathbb{F}_{256}, v = 90$)	0.30	0.50
Rainbow($\mathbb{F}_{16}, v_1 = 19$)	0.71	0.92
Rainbow($\mathbb{F}_{31}, v_1 = 17$)	0.43	0.68
Rainbow($\mathbb{F}_{256}, v_1 = 20$)	0.075	0.14

Table 3: Probability for coincidences in random variables for different fields for UOV and Rainbow.

C Probability for Equal Random Variables

For a field with q elements and a number of v_1 vinegar variables, the event that two randomly generated sets of v_1 vinegar variables have at least one coincidence is the complementary event of no coincidences at all. The probability for this is $p_1 = 1 - \left(\frac{q-1}{q}\right)^{v_1}$. If we compute this for a number of $i \leq k$ sets of random values, then we get the probability that at least in one comparison at least one coincidence occurs by $p_2 = 1 - (1 - p_1)^i$. If we use common parameters for UOV and Rainbow schemes, we see that such occurrences are quite frequent.

As we can see in Table 3, it is quite probable that one or more variables have a value in common with older sets of variables. So one should not deny any signature where a coincidence occurs, but define a threshold value.