Handling Vinegar Variables to Shorten Rainbow Key Pairs

Gustavo Zambonin(✉), Matheus S. P. Bittencourt, and Ricardo Custódio

Departamento de Informática e Estatística,
Universidade Federal de Santa Catarina,
Florianópolis 88040-900, Brazil
gustavo.zambonin@posgrad.ufsc.br, matheus.spb@grad.ufsc.br,
ricardo.custodio@ufsc.br

Abstract. Multivariate quadratic equations are the basis of one of the main mathematical techniques for the creation of digital signatures that are quantum-resistant. In these schemes, the creation and verification of signatures is highly efficient. However, key sizes are quite impractical and orders of magnitude greater than conventional schemes. One of the best-known signature schemes built upon multivariate equations is called Rainbow, which is based on the Oil-Vinegar principle. We observe that the reuse of vinegar variables in the signature generation step of the Rainbow scheme leads to a shorter representation of its central map, and thus, of the entire private key. We analyse the security implications of this strategy and present a modification to the Rainbow scheme, enabling a private key size reduction of up to 85% with secure parameters. Additionally, this framework can be applied on top of already existing schemes that shorten either private or public keys, spawning derivatives that reduce the total key pair size by a factor of 3.5.

Keywords: Multivariate cryptography · Digital signatures · Rainbow

1 Introduction

Secure exchange of messages is nowadays treated as a requirement in digital systems, instead of a privilege. It is often mandatory that data is not altered in transit, that its sender is uniquely identifiable and that it cannot deny having sent the message. These notions, known as integrity, authenticity and non-repudiation, are achieved through the use of cryptographic foundations known as digital signatures. Data protected with such a method is adequate to prevent forgery and ensure confidentiality, according to Goldreich [13].

Conventional digital signature schemes are predominantly bound to one of two mathematical problems, namely integer factorisation and discrete logarithm. The most common examples are the RSA and ECDSA signature schemes [12], respectively. Nonetheless, in the wake of possible quantum adversaries, these problems are provably solvable in polynomial time, due to Shor's

J. Buchmann et al. (Eds.): AFRICACRYPT 2019, LNCS 11627, pp. 391–408, 2019.
https://doi.org/10.1007/978-3-030-23696-0_20

algorithm [26]. Ergo, the design of quantum-resistant, or post-quantum digital signature schemes, is indispensable to preserve secure communications in a scenario with quantum computers.

The creation of post-quantum digital signatures can be achieved through several approaches, one of which is based on systems of multivariate quadratic equations. Due to this fact, it is named multivariate cryptography, and schemes derived from this mathematical foundation are based on problems not known to be more efficiently solved by quantum computers [2]. Moreover, their signature generation and verification procedures are extremely efficient [7], since most computations rely only on simple finite field arithmetic.

It is known that multivariate cryptography hosts distinct schemes with several combinations of security parameters, signature and key pair lengths, as summarised by the authors of [8]. A balanced choice lies in the Rainbow signature scheme [9], itself a generalisation of the classic Unbalanced Oil and Vinegar (UOV) scheme [16]. It is a popular scheme, with several improvements featured in the literature, and multiple hardware implementations, *e.g.* [5,27,34]. Furthermore, it is currently featured in the second round of the standardisation process organised by the National Institute of Standards and Technology (NIST) [1].

One major drawback of multivariate cryptography, including Rainbow, is the size of private and public keys. While conventional signature schemes have key sizes that are a few bytes long, schemes based on multivariate equations feature keys that are dozens of kilobytes long. Hence, it is desired to reduce these by means of novel mathematical strategies, without decreasing the security of the scheme. Various strategies are applied to shorten keys, such as generating systems of equations represented by sparse matrices, or elements produced by cyclic recurrences. However, the security implications of such modifications are often obscure and possibly harmful.

Our Contributions. We present a general framework that can be applied to any Rainbow-like signature scheme, with the final intent of reducing private key sizes. It manipulates vinegar variables that are originally chosen randomly to successfully invert the central map. These variables are now locked into the private key, thus reducing the degree of all monomials that feature such variables, lowering the total number of field elements used to represent it by up to a factor of 6.25. To sustain our proposal, we analyse the relation between signatures and the choice of vinegar variables, security implications of this strategy and experiment on known Rainbow variants. To the best of our knowledge, Rainbow variants proposed in the literature allow the reduction of private or public keys, but not both simultaneously. We show that our proposal allows for a shorter private key without preventing modifications to the public key. Thus, by making use of known proposals to reduce public keys, we create the first Rainbow variants that reduce the total size of the key pair.

Notation. We will use the following symbols throughout this work. The symbol $\xleftarrow{\$}$ is read as "chosen randomly from", and $\approx_\varepsilon$ means that two numbers are equal within a precision of ε. A finite field $\mathbb{F}$ with order q and elements as vectors of length n is represented as $\mathbb{F}_q^n$, with q and n omitted for brevity if appropriate. The cardinality of a set S is given by $|S|$. This notation may also be used as the absolute value of an integer, if applicable. The usual function composition is given by the symbol $\circ$, and the inverse of a function f is given by f^{-1}. The usual standard deviation and mean functions for a set of elements S are respectively given by $\sigma(S)$ and $\mu(S)$.

Organisation. The next sections are organised as follows. Section 2 succinctly describes the theoretical background needed to assimilate our proposal, with a definition of the Rainbow signature scheme in Subsect. 2.1 and a review of works that already reduce keys for this scheme in Subsect. 2.2. Section 3 presents the rationale for our proposal and a formal description, alongside a security analysis. Section 4 shows the impact of our proposal when applied to the original Rainbow and variants. Finally, Sect. 5 offers our final considerations.

2 Preliminaries

2.1 Original Rainbow Signature Scheme

We will present below a description of the Rainbow signature scheme, a generalised version of the UOV scheme that reduces the length of keys and signatures. It consists of several "oil and vinegar" layers, that are combined to create a "rainbow". Consider a finite field $\mathbb{F}_q$ and $u, n \in \mathbb{N}$ where $u \leq n$. Choose a sequence of integers $v_1, \ldots, v_u$ such that $0 = v_0 < v_1 < \cdots < v_u < v_{u+1} = n$. Take the usual set $V = \{1, \ldots, n\}$ and define the vinegar variables as $V_l = \{1, \ldots, v_l\}$ for all $l \in \{1, \ldots, u\}$. Observe that $v_l = |V_l|$ and $V_1 \subset \cdots \subset V_u = V$. Oil variables are given by $O_l = \{v_l + 1, \ldots, v_{l+1}\}$. Note that $o_l = |O_l|$ and $O_l = V_{l+1} - V_l$. Let $m = n - v_1$. Now, we define vector spaces spanned by quadratic Oil-Vinegar polynomials of the form

$$P_l = \sum_{i,j \in V_l} \alpha_{ij} \cdot x_i \cdot x_j + \sum_{i \in V_l, j \in O_l} \beta_{ij} \cdot x_i \cdot x_j + \sum_{i \in V_l \cup O_l} \gamma_i \cdot x_i + \delta. \tag{1}$$

Key Generation. The central map of Rainbow is defined as $\mathcal{F} : \mathbb{F}^n \longrightarrow \mathbb{F}^m$, with the following construction: for each layer l, $F_l = (F_l^1, \ldots, F_l^{o_l}) \xleftarrow{\$} P_l$, and $\mathcal{F} = (F_1, \ldots, F_l)$. Since each sequence of vinegar variables in a layer contains all variables from the previous layer, this allows for the inversion of this map. Further, let $\mathcal{S} : \mathbb{F}^m \longrightarrow \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \longrightarrow \mathbb{F}^n$ be two affine invertible maps, used as the trapdoor to this construction. Let $\mathcal{P} : \mathbb{F}^n \longrightarrow \mathbb{F}^m$ as $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$. Coefficients $\alpha_{ij}, \beta_{ij}, \gamma_i, \delta \in \mathbb{F}$ are chosen randomly. The private key is the triple $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ and the public key is the map $\mathcal{P}$.

Signature Generation. To sign a message M, consider a cryptographic hash function $\mathcal{H} : \{0,1\}^* \longrightarrow \mathbb{F}^m$, and obtain the message digest $d = \mathcal{H}(M)$. The signature will be the set of variables which yield the solution to the equation $\mathcal{P}(x_1, \ldots, x_n) = d$. Compute $x = \mathcal{S}^{-1}(d)$. To generate $y = \mathcal{F}^{-1}(x)$, every layer must be inverted recursively. Start by randomly choosing values for $x_1, \ldots, x_{v_1}$ and inserting them into the first layer. This will bring forth a system of o_1 linear equations in $x_{v_1+1}, \ldots, x_{v_2}$. It can be solved with an algorithm such as Gaussian elimination. If the system does not have a solution, new vinegar variables have to be chosen. These solutions can then be substituted into the next layer, which will create a system of o_2 linear equations, that can be solved analogously. This procedure is repeated until all layers are solved. Finally, we compute $\sigma = \mathcal{T}^{-1}(y)$.

Signature Verification. To verify a signature, compute $d' = \mathcal{P}(\sigma)$. If $d = d'$, then the signature is valid, and invalid otherwise.

Finally, denote an instance of the scheme by Rainbow $(\mathbb{F}_q, v_1, o_1, \ldots, o_u)$. Note that when $u = 1$, we get the UOV scheme. Measured in field elements, the size of a private key is

$$|\mathcal{K}_{Pr}| = m^2 + m + n^2 + n + \sum_{k=1}^{u} o_k \cdot \left(\frac{v_k \cdot (v_k + 1)}{2} + v_k \cdot o_k + v_{k+1} + 1 \right), \quad (2)$$

whereas the size of a public key is

$$|\mathcal{K}_{Pu}| = m \cdot \frac{(n+1) \cdot (n+2)}{2}. \quad (3)$$

Further details on the construction of Rainbow may be found on [7, Section 3.3].

2.2 Related Works

Schemes based on multivariate cryptography with modifications that enable the reduction of private key sizes have been suggested even before Rainbow was created. Tame transformation schemes, such as the ones listed by Wolf and Preneel in [29], feature sparseness in their maps, a common strategy used to shorten private keys. However, these schemes were either broken, as summarised by the authors in [10], or in the case of Enhanced TTS, new parameters were suggested, and it was subsequently found to be a special case of Rainbow [28].

Additionally, there have been several published variations of Rainbow with the same goal, making use of distinct approaches. A scheme called Lite-Rainbow-0 [25] makes use of a small pseudorandom number generator (PRNG) seed to replace the private key entirely. This shortens the private key by a factor of approximately 99.8%, but greatly increases the cost for signature generation. NC-Rainbow was proposed in [31] with a novel strategy based in non-commutative rings to reduce a private key by up to 75%. However, it was shown by independent researchers to be insecure [14,28]. Other variants called MB-Rainbow [30] and NT-Rainbow [33] employ sparseness of maps to reduce the number of terms in the private key by up to 40%.

The authors merged MB- and NC-Rainbow into a single scheme called MNT-Rainbow [32], shortening private keys by up to 76%. Nevertheless, the original schemes were deemed insecure and new parameters were suggested in [18]. It also proposes a new scheme called Circulant Rainbow, which reduces the private key by up to 45% due to the concept of rotating relations. Yet, it was broken shortly after [15].

It is also relevant to cite the approach by the authors of [21], which is, to the best of our knowledge, the main method for public key reduction without compromises to the signature size. It is summarised in several publications [20,22,24]. However, these cannot be combined with the private key improvements previously cited. Furthermore, it appears that the introduction of structures in the private key is highly threatening to the overall security of a Rainbow scheme. We will subsequently present a novel approach to these issues.

3 Our Proposal

We will describe our improvement to Rainbow-like signature schemes below, as well as supporting research on its soundness. Subsections 3.1 and 3.2 give a formal description of our modifications. In Subsect. 3.3, we look into the probability of matrices with elements in finite fields being invertible. In Subsect. 3.4, we present a statistical analysis of the structure of signatures created by our method, and finish with a security overview in Subsect. 3.5.

3.1 Modification to the Original Scheme

Our approach consists of modifications to the key and signature generation steps of Rainbow-like signature schemes. We propose to reuse the first set of vinegar variables for several signatures and replace these only when necessary, *i.e.* situations where the central map cannot be inverted and creating a signature would fail. By locking such variables and substituting them on the central map $\mathcal{F}$ early in the key generation algorithm, we create a $\mathcal{F}'$ linear in v_1, thus reducing storage requirements. This approach does not modify the underlying structure of the private key, but rather of the central map preimages.

To induce lower storage requirements for key pairs of Rainbow-like schemes, we explore constructions given in the literature and suggest general alterations to use our proposal. As per Subsect. 2.2, most variants that shorten private keys are structural in nature, that is, the key space is limited by some heuristic with the intent of producing a compact private key. Moreover, the main approach to reduce public keys [19] prevents alterations to the private key, since it indirectly generates $\mathcal{F}$ from a partial public key through linear relations between the maps.

This division of improvements is blurred by our proposal. We present general methods based on different techniques that shorten private keys in all Rainbow-like schemes. We collectively denote these by Rainbow-η and use the same definitions as in Subsect. 2.1, further denoting the vinegar variables for the first layer as $\widetilde{V}_1 = (x_1, \ldots, x_{v_1})$.

Rainbow-η_1 Key Generation. We use the fact that a PRNG has the ability to regenerate the same sequence of numbers given a seed. The choice of such a generator is outside the scope of our work, and we assume that a cryptographically secure PRNG is chosen. This approach is similar to Lite-Rainbow-0, but it is not as costly, since the private key does not need to be regenerated before every signature generation. It is best suited to environments in which an efficient generator is previously supplied.

We bound the creation of the key pair to a seed $\mathbf{S}$. We are not aware of any Rainbow variants that disallow this practice. Thus, $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$, as well as the public key $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$ are generated through the target scheme key generation algorithm, seeded by $\mathbf{S}$. We set $\widetilde{V}_1 \xleftarrow{\$} \mathbb{F}$, and substitute these into $\mathcal{F}$, giving $\mathcal{F}'$. According to Subsect. 3.3, in the rare case that a failure occurs in the central map inversion algorithm, we use $\mathbf{S}$ to regenerate $\mathcal{F}$, choose other values for $\widetilde{V}_1$ and create a different $\mathcal{F}'$. The private key of Rainbow-η_1 is $(\mathbf{S}, \mathcal{S}, \mathcal{F}', \mathcal{T})$ and the public key is $\mathcal{P}$.

Rainbow-η_2 Key Generation. This approach is based on the fact that a private key owner is able to recover the original $\mathcal{F}$ through the possession of all other private maps and the public key. We make use of the linear relations given by the authors of [21] and applied in the definition of the well-known CyclicRainbow scheme. A short explanation is given below, with the full rationale available in [19, Chapter 7].

Consider the public key $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$ and let $\mathcal{Q} = \mathcal{F} \circ \mathcal{T}$. Denote $\widetilde{Q}$ as a matrix containing only coefficients of the quadratic monomials from $\mathcal{Q}$, and define $\widetilde{F}$ and $\widetilde{P}$ similarly. Further let $\widetilde{T}$ be the matrix representation of $\mathcal{T}$, with its coefficients $t_{ij}, i, j \in \{1, \ldots, n\}$, and define $\widetilde{S}$ analogously. By fixing t_{ij}, the composition of $\mathcal{P}$ actually represents a linear relation between coefficients q_{ij}^k, f_{ij}^k of the monomial $x_i \cdot x_j$ in the k-th component of, respectively, $\mathcal{Q}$ and $\mathcal{F}$, with the form

$$q_{ij}^k = \sum_{r=1}^{n} \sum_{s=r}^{n} \alpha_{ij}^{rs} \cdot f_{rs}^k, \quad \alpha_{ij}^{rs} = \begin{cases} t_{ri} \cdot t_{si} & \text{if } i = j, \\ t_{ri} \cdot t_{sj} + t_{rj} \cdot t_{si} & \text{otherwise,} \end{cases} \qquad k \in \{v_1 + 1, \ldots, n\}. \tag{4}$$

This can be simplified, since $\mathcal{F}$ does not allow quadratic monomials with only oil variables, and results in

$$q_{ij}^k = \sum_{r=1}^{v_l} \sum_{s=r}^{v_{l+1}} \alpha_{ij}^{rs} \cdot f_{rs}^k, \quad k \in O_l, \quad l \in \{1, \ldots, u\}. \tag{5}$$

A square matrix of order $\frac{n^2+n}{2}$ is created to further streamline the previous equations. Given a particular monomial ordering, let $A = (\alpha_{ij}^{rs})$ such that $i, j, r, s \in \{1, \ldots, n\}$, where $i \leq j$ and $r \leq s$ denote row and column indices, respectively. Thus, we have that $\widetilde{P} = \widetilde{S} \cdot \widetilde{Q}$ and $\widetilde{Q} = \widetilde{F} \cdot A^{\mathrm{T}}$. We note that the performance of this method is lower than that of Rainbow-η_1. However, it is a general technique that works on all Rainbow-like schemes.

Observe that the central map may not feature any linear or constant terms, due to the use of the above relations. This does not lower the overall security of the scheme, due to the fact that they are not multiplied with quadratic terms. With this implication in mind, the usual key generation algorithm for the target scheme is employed, yielding $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ and $\mathcal{P}$ at a marginally faster rate. Substitute the sequence $\widetilde{V}_1 \xleftarrow{\$} \mathbb{F}$ into $\mathcal{F}$, giving $\mathcal{F}'$. By the relations above, one is able to reconstruct $\mathcal{F}$ with no additional mechanisms if the central map inversion algorithm fails. The private key of Rainbow-η_2 is $(\mathcal{S}, \mathcal{F}', \mathcal{T})$ and the public key is $\mathcal{P}$.

Signature Generation. A digest $d = \mathcal{H}(M)$ from a message M is signed with a similar procedure. Compute $x = \mathcal{S}^{-1}(d)$, and attempt to generate $y = \mathcal{F}'^{-1}(x)$ by inverting every layer recursively. The first layer already has $\widetilde{V}_1$ set, and the remaining linear system needs only to be solved by providing appropriate values of d. It will generate a new set of vinegar variables, that can be used on the next layer, until all layers are solved. If any of the transitory systems are not solvable, a new $\widetilde{V}_1$ is chosen and $\mathcal{F}'$ regenerated, according to one of the methods given above. We finish by computing $\sigma = \mathcal{T}^{-1}(y)$.

Signature Verification. This step does not change. If $d = \mathcal{P}(\sigma)$, then the signature is valid, and invalid otherwise.

By making the first layer linear and substituting the remaining variables, the size of the private key is now

$$\begin{aligned}|\mathcal{K}^{\eta}_{Pr}| &= m^2 + m + n^2 + n + |\widetilde{V}_1| \\ &\quad + \sum_{k=1}^{u} o_k \cdot \left(\frac{(v_k - v_1)(v_k - v_1 + 1)}{2} + (v_k - v_1) \cdot o_k + (v_{k+1} - v_1) + 1 \right), \end{aligned} \tag{6}$$

plus the additional size of $\mathbf{S}$ if the Rainbow-η_1 method is used. One needs to store $\widetilde{V}_1$, since it is part of the central map preimage, used on further map applications. The public key size does not change.

3.2 Application to the EF-CMA Variant

The Rainbow submission to the NIST standardisation process [6] presents a scheme description that diverges from the original works. The authors introduce modifications that provide security against the existential forgery under chosen-message attack (EF-CMA) model, whereas the original scheme only offers security against universal forgery. These changes are built upon the introduction of a random salt. We will briefly describe this approach, with the intent of preventing the recalculation of $\mathcal{F}'$ in the case that $\widetilde{V}_1$ is not suitable. Let us denote this method as **Rainbow-η_3**.

Key Generation. Consider $w \in \mathbb{N}$ as the length of the aforementioned salt. Generate private and public keys as per Subsect. 3.1. The private key for this scheme is $(\mathcal{S}, \mathcal{F}', \mathcal{T}, w)$, with the addition of $\mathbf{S}$ in the case of Rainbow-η_1. The public key is $(\mathcal{P}, w)$.

Signature Generation. Let $r \xleftarrow{\$} \{0,1\}^w$. The digest value is calculated as $\mathbf{d} = \mathcal{H}(\mathcal{H}(M) \mid\mid r)$, where M is the message. The value $x = \mathcal{S}^{-1}(\mathbf{d})$ is obtained as usual. In the rare case that the $y = \mathcal{F}'^{-1}(x)$ preimage calculation does not succeed, new variables in $\widetilde{V}_1$ are chosen. However, the addition of a random salt to the original message digest alters $\mathbf{d}$ completely, due to the cryptographic hash function application. Thus, it is only necessary to generate a new r and restart the signature generation process, such that $\widetilde{V}_1$, and consequently $\mathcal{F}'$, are not modified. Alternatively, if the preimage is generated successfully, we finish by letting $z = \mathcal{T}^{-1}(y)$ and $\sigma = (z, r)$.

Signature Verification. Recalculate the digest value $\mathbf{d}$. If $\mathbf{d} = \mathcal{P}(z)$, the signature is valid, and invalid otherwise.

The size of the private and public keys increase in exactly one element due to the addition of w. Real implementations of Rainbow-η_3 are tested on Sect. 4.

3.3 Invertibility of $\mathcal{F}$

Recall that, to create a Rainbow signature, the central map $\mathcal{F}$ needs to be inverted. Random guessing of vinegar variables is done in order to create a solvable linear system. It is also known that the central map is expressed as multivariate systems of equations, which can be themselves interpreted as multidimensional matrices of coefficients. Observe that, to describe these in a clearer way, a given monomial ordering is used such that only usual matrices are needed. With this in mind, we first derive the probability that a random matrix with elements in $\mathbb{F}$ is invertible.

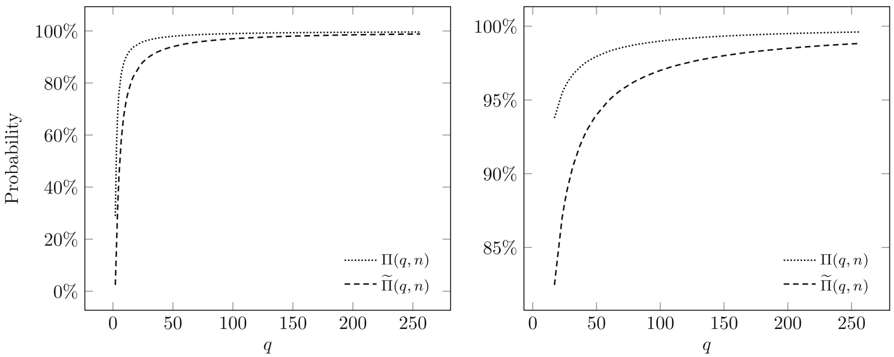
Assume a square matrix M of order n such that $m_{ij} \in \mathbb{F}_q, i, j \in \{1, \ldots, n\}$. For M to be invertible, it must be composed entirely of vectors, *i.e.* its rows $m_i \in \mathbb{F}^n$, that are linearly independent. The zero vector $(0, \ldots, 0) \in \mathbb{F}^n$ is linearly dependent of all other vectors. Thus, $m_1 \neq (0, \ldots, 0)$, with all other $q^n - 1$ possible vectors eligible. m_2 must not feature any of the q multiples of m_1, and $q^n - q$ vectors remain. Without loss of generality, $m_k \neq c_1 v_1 + c_2 v_2 + \cdots + c_{k-1} v_{k-1}, c_k \in \mathbb{F}$, and $q^n - q^{k-1}$ vectors can be selected. Then, the probability that all vectors chosen are linearly independent is

$$\begin{aligned} \Pi(q, n) &= \prod_{k=1}^{n} \frac{q^n - q^{k-1}}{q^{-n}} \\ &= \prod_{k=1}^{n} 1 - q^{-k}. \end{aligned} \tag{7}$$

In the context of Rainbow, the number of layers directly influences $\Pi(q,n)$, since it dictates how many linear systems have to be solved. In other words, all square matrices of size $v_i, i \in \{1,\ldots,u\}$ need to be invertible to achieve a preimage under $\mathcal{F}$. Thus, the probability

$$\Pi(q,n,u) = \prod_{i=1}^{u}\prod_{k=1}^{v_{i+1}} 1 - q^{-k} \tag{8}$$

more accurately represents the upper bound for these chances. In the literature, the usual number of layers for a Rainbow instance is two, and we will denote this common case as $\Pi(q,n,2) = \widetilde{\Pi}(q,n)$. Note that $\Pi(q,n,1) = \Pi(q,n)$. Hence, schemes with more layers have a slightly lower probability of success in the signature generation preimage step.



(a) $\mathrm{argmin}_{56\leq n\leq 90}$ of $\Pi(q,n)$ and $\widetilde{\Pi}(q,n)$.

(b) Figure 1a, with $q \geq 16$.

Fig. 1. Probability of obtaining an invertible matrix, populated with field elements where $q \in \{2,\ldots 256\}$ and q is a prime power, given the quantity of layers of Rainbow.

Parameters for Rainbow are selected according to a number of restrictions, imposed by attacks that may harm the security of the scheme. Furthermore, note that the central map can be represented as square matrices of order n. Hence, we choose $n \in \{56,\ldots,90\}$ from [19, Tables 6.4, 6.8, 6.13] and calculate the probability that a random matrix is invertible in finite fields of typical orders. For instance, $\Pi(16,90) \approx 93.3594\%$ and $\Pi(256,56) \approx 99.6078\%$. Figure 1 depicts the lowest probabilities computed for the appropriate range. To simulate layering, we set $v_i = i \cdot \lceil \frac{n}{u} \rceil$ and approximate to n when needed.

It is also useful to calculate $\lim_{n\to\infty} \Pi(q,n)$ to observe changes in the probability with the growth of m. Note that this is very similar to the Euler function $\phi(q)$. Ergo, we can use one of Euler's identities to redefine the above limit as

$$\Pi(q) = \sum_{k=-\infty}^{\infty} (-1)^k q^{\frac{-3\cdot k^2+k}{2}} \tag{9}$$

and obtain a fast approximation of the probability when n tends to infinity. We use the SageMath language arbitrary precision real numbers to obtain these values and find out that, when $n \geq 56$, $\Pi(q) \approx_{10^{-18}} \Pi(q,n)$ and $\widetilde{\Pi}(q) \approx_{10^{-8}} \widetilde{\Pi}(q,n)$. Thus, Fig. 1 also accurately reflects the behaviour of $\Pi(q)$, *i.e.*, current values of n already reach effective upper bounds for this probability.

If we consider that the two-dimensional coefficient matrix of $\mathcal{F}$ has an effective size of $\frac{n^2+n}{2}$ due to the aforementioned monomial ordering strategy, we note that the inversion event happens almost surely. This evidence shows that computing a preimage in order to sign a message happens at the first try with high probability in a wide range of Rainbow configurations. Therefore, the cost of a central map reconfiguration, in the case that chosen vinegar variables do not lead to an invertible central map, is amortised by the overwhelming probability that a signature is successfully generated.

3.4 Similarity of Multiple Signatures

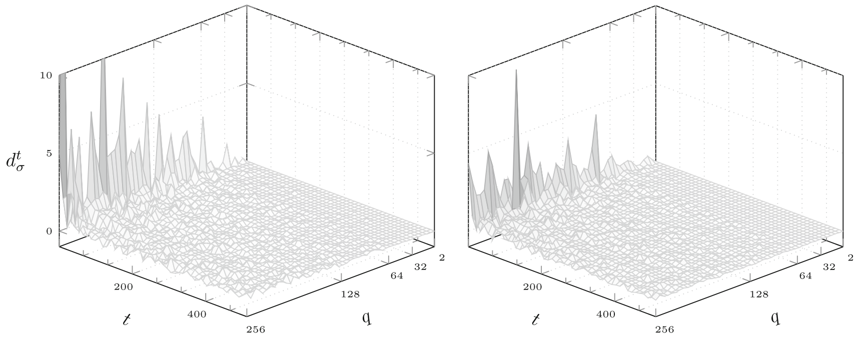
Vinegar variables chosen to invert the central map are an integral part of the preimage $y = \mathcal{F}^{-1}(x)$. For instance, in the case $u = 2$, these make roughly a third of the output, considering common parameters for Rainbow. Further, recall that there are approximately q^v possibilities for y. Our proposal eliminates this choice by locking vinegar variables into the private key. Hence, it is essential to know if such variables create patterns in which private information may leak through a multi-target attack. We use the SageMath PRNG, which implements a front-end to the `/dev/urandom` Linux kernel space generator.

Recall that a message digest d is signed instead of the entire document. Evidently, a secure cryptographic hash function shall produce an output that appears to be random. The application $x = \mathcal{S}^{-1}(d)$ does not affect this behaviour, since the map is also random. Hence, we need not simulate this calculation in this analysis. According to Subsect. 3.3, the inversion $y = \mathcal{F}^{-1}(x)$ creates a valid preimage with overwhelming probability, where the first v_1 elements of any y will be the same.

We observe the distribution of field elements in vectors after the final function application, that is, $z = \mathcal{T}^{-1}(y)$. Let $Z'_t = (z_1, \ldots, z_t) \xleftarrow{\$} \mathbb{F}^n, t \in \mathbb{N}$ be a t-uple of “signatures”. We build the sequence $Z_t = (z_1^1, z_1^2, \ldots, z_1^n, z_2^1, \ldots, z_t^{n-1}, z_t^n)$. When part of the vector y is fixed, we will instead denote these by $\widetilde{Z}'_t$ and $\widetilde{Z}_t$. Our hypothesis is that Z_t and $\widetilde{Z}_t$ will behave similarly to observations sampled from the discrete uniform distribution $\mathcal{U}\{0, q-1\}$. It is known that its standard deviation, where r values are observed in an equally likely manner, is equal to $\sqrt{\frac{r^2-1}{12}}$. For a finite field $\mathbb{F}$, we set $r = q$ and obtain the desired value. It is expected that

$$\lim_{t\to\infty} \sigma(\widetilde{Z}_t) = \sqrt{\frac{q^2-1}{12}}, \tag{10}$$

suggesting that greater values of n and t approximate faster to the theorised standard deviation.



(a) d_σ^t for $n = 42, v_1 = 17$.

(b) d_σ^t for $n = 90, v_1 = 35$.

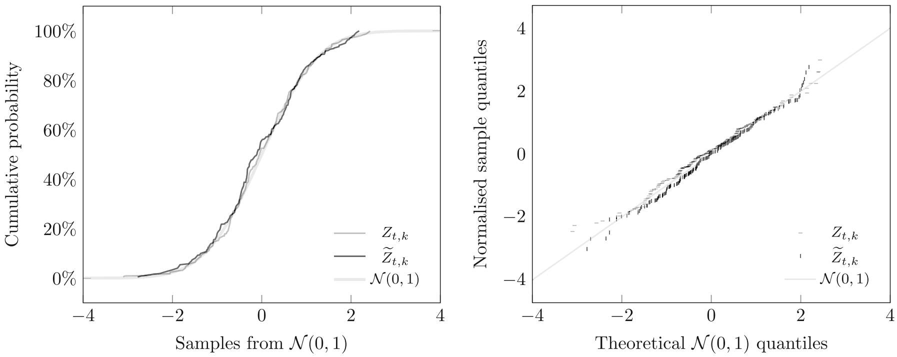
Fig. 2. Difference of standard deviations when $t \in \{1, \dots, 1024\}$, and $q \in \{2, \dots, 256\}$ with q as a prime power.

Let us denote the absolute difference between standard deviations for a value t as $d_\sigma^t = |\sigma(Z_t) - \sigma(\widetilde{Z}_t)|$. Figure 2 shows the amplitude of such values for various values of q and t. We note that the largest values of d_σ^t occur for finite fields of higher orders and lower t. For instance, given the finite field $\mathbb{F}_{223}^{42}$, we have $d_\sigma^1 \approx 8.25$, and for a slightly higher t, we obtain a much lower value $d_\sigma^{11} \approx 0.24$. This behaviour is also observed within absolute differences of means, defined analogously as d_μ^t. The field $\mathbb{F}_{191}^{42}$ gives the values $d_\mu^1 \approx 5.31$ and, comparatively, $d_\mu^9 \approx 1.14$.

The comparison of expected and obtained standard deviations and means in our experiments, gives positive results and confirms the law of large numbers. Still, it is interesting to look at the diffusion of values within $\widetilde{Z}_t$ and infer that it does not simply simulate the mean and standard deviation for a known discrete uniform. We count the amount of values for each class $k \in \{0, \dots, q-1\}$ and refer to them by $Z_{t,k}$ and $\widetilde{Z}_{t,k}$. By the central limit theorem, these counts should be normally distributed.

Figure 3 shows the cumulative distribution function (CDF) plot and the Q-Q (quantile-quantile) plot for such samples. The expected CDF, as well as examples for Z_t and $\widetilde{Z}_t$, show that all values are fairly distributed, with small variations due to the random generation of field elements. However, we note that this is due to the low number of classes, *i.e.* the order of the finite field, and experimentally confirm that such discrepancies are largely reduced with $q = 2^{10}$. This is further confirmed by the Q-Q plot created with rankits, where the points are sufficiently close to the $y = x$ expected line.

Our argument indicates that, even if part of the preimage created by the central map is fixed, the remaining affine map application disrupts this pattern with high efficacy. Hence, an attacker with possession of multiple signatures created by our method would not be more capable of forging a new signature or deducing private information.



(a) CDF plot.

(b) Q-Q plot.

Fig. 3. Distribution of counts of elements in Z_t and $\widetilde{Z}_t$ such that $t = 2^{16}$ for $\mathbb{F}_{256}^{90}$.

3.5 Security Analysis

A variety of attacks currently thwart the security of Rainbow-like signature schemes if parameters are not chosen carefully. We will briefly state each of those, along with their estimated complexities [23], and argue that our methods do not facilitate such attacks.

Direct Attack. An attacker with possession of a digest d and the public key $\mathcal{P}$ tries to solve $\mathcal{P}(x) = d$. This is done by fixing some of the variables and applying an algorithm built upon the theory of Gröbner basis, such as the Hybrid approach [3]. While it is hard to pinpoint the exact running time of such methods, the authors give an estimation of its asymptotic complexity in Eq. 5 of the aforementioned work.

UOV Attack. The multi-layer approach of Rainbow does not hinder attacks that also work on the UOV signature scheme. This attack was originally created by Kipnis and Shamir [17] to break the Balanced Oil-Vinegar scheme. The objective of this attack is to obtain an equivalent private key by means of finding the preimage of a specific oil subspace under the map $\mathcal{T}$. The complexity of the generalised attack for unbalanced schemes [16] is $o_u^4 \cdot q^{n-1-2\cdot o_u}$ field multiplications.

MinRank Attack. All systems of polynomials in the public key $\mathcal{P}$ may be individually represented as matrices. This attack consists in finding linear combinations of these, such that they have a lesser rank than v_2, in the case of Rainbow. This allows an attacker to isolate the central map polynomials from the first layer of Rainbow, and analogously recover the remaining layers with a much lower effort. In the context of Rainbow [4], its complexity is $q^{v_1+1} \cdot m \cdot (\frac{n^2}{2} - \frac{m^2}{6})$ field multiplications.

HighRank Attack. In a similar way to MinRank, linear combinations of public key matrices are used to find the variables which appear the lowest number of times in the central map. This is used to identify the last Rainbow layer, and obtain the previous layers similarly. The complexity of the improved attack [11] is $q^{o_u} \cdot \frac{n^3}{6}$ field multiplications.

Rainbow-Band-Separation Attack. An extension of the UOV-Reconciliation attack by the same authors [11] that targets Rainbow, with the intent of producing an equivalent private key. It explores the fact that the central map matrix representation is composed of zeroes on its lower right corner. These yield quadratic equations which, if solved, lead to an alternative private key. The complexity of this attack is given by the hardness of solving a large system of equations, as seen above, is hard to estimate.

Side-Channel Attacks. It may be observed that none of the proposed Rainbow variants, as well as the original scheme, present constant time signature generation algorithms. Particularly, in Rainbow-η_2, a considerable amount of computation is added to the signature algorithm when one of the systems is not solvable. In a chosen message attack, one may observe the time spent on multiple signature generation steps and easily check if the linear systems are solvable, thus obtaining information about the central map. Although there are no known attacks that make use of this technique, it is possible that there may exist information leaks when applying our methods to Rainbow-like schemes.

We do not discard the possibility that specialised attacks exist, particularly ones that take in account multiple signatures, due to our fixing of vinegar variables. However, we have seen in Subsect. 3.4 that signatures generated by our method are comparably random with respect to conventional Rainbow signatures. Furthermore, we note that most attacks look for special structures within the private key. While our methods indeed modify the private key representation, it is still present in its entirety on the public key composition, which is the only information available to malicious entities that can be possibly used to forge signatures. We thus suggest that the right choice of parameters is made whenever our methods are applied, *e.g.* according to [23], to protect the scheme instance against these attacks.

4 Enhancement of Existing Schemes

Our method does not depend on special structures inserted on the private key. Consequently, it can be applied to all known Rainbow-like schemes. We experiment with several sets of parameters and observe the reduction of private keys. It is known that there are various limitations for the choice of parameters that lead to secure instances of Rainbow [23]. We implement several known guidelines and confirm that our proposal does indeed work for a large range of parameters. However, we only show results for known secure parameter sets to prevent accidental endorsement of untested, and possibly insecure, instances.

Table 1. Reduction of Rainbow key sizes, in bytes, for various instances of the scheme.

Instance	Parameters	n	m	$\lvert\mathcal{K}_{Pr}\rvert$	$\lvert\mathcal{K}^{\eta}_{Pr}\rvert$	Difference
I-a	$(\mathbb{F}_{16}, 32, 32, 32)$	96	64	100208	33152	-66.92%
I-b	$(\mathbb{F}_{31}, 36, 28, 28)$	92	56	114308	31676	-72.29%
I-c	$(\mathbb{F}_{256}, 40, 24, 24)$	88	48	143384	33024	-76.97%
III-b	$(\mathbb{F}_{31}, 64, 32, 48)$	144	80	409463	87628	-78.60%
III-c	$(\mathbb{F}_{256}, 68, 36, 36)$	140	72	537780	99656	-81.47%
IV-a	$(\mathbb{F}_{16}, 56, 48, 48)$	152	96	376140	103336	-72.53%
V-c	$(\mathbb{F}_{256}, 92, 48, 48)$	188	96	1274316	218984	-82.82%
VI-a	$(\mathbb{F}_{16}, 76, 64, 64)$	204	128	892078	233044	-73.88%
VI-b	$(\mathbb{F}_{31}, 84, 56, 56)$	196	112	1016868	217244	-78.64%
P-080	$(\mathbb{F}_{256}, 17, 17, 9)$	43	26	19208	5914	-69.21%
P-100	$(\mathbb{F}_{256}, 26, 22, 21)$	69	43	75440	23193	-69.26%
P-128	$(\mathbb{F}_{256}, 36, 28, 15)$	79	43	103704	22110	-78.68%
P-192	$(\mathbb{F}_{256}, 63, 46, 22)$	131	68	440638	71773	-83.71%
P-256	$(\mathbb{F}_{256}, 85, 63, 30)$	178	93	1086971	164721	-84.85%

We show results for the application of our method in Table 1, considering the following Rainbow instances. Conservative choices were made by the Rainbow submission authors [6] to fit security categories as requested by NIST. We apply our method to these recent proposals, and additionally choose parameters from Petzoldt [19, Table 6.12] for further comparison. The latter are named P-ℓ, where ℓ is the security level in bits. Indeed, the choice of v_1 remarkably affects the results. Moreover, a minimal value of o_u is also known to further reduce the private key size. Indeed, we suggest that $v_1 \geq o_u$ as much as possible to maximise the results of our method. However, we remark that one must set sufficient parameters for o_i such that the scheme still resists direct and UOV attacks.

The case of Rainbow variants is slightly more convoluted. Schemes claim optimisations of the private key often through the inclusion of inner structuring. To measure the impact of our method within the context of these schemes, it is imperative to understand such structures. For instance, it may be the case that a method introduces sparseness related to specific vinegar variables. Thus, the reduction would not be equally distributed over the private key elements and, as such, our method would have its efficiency reduced.

To the best of our knowledge, the schemes presented in Subsect. 2.2 feature changes that target the whole private key evenly. Hence, our method would yield similar results to those in Table 1 if this assumption is true. However, it is also the case that some variants were subsequently broken or new parameters were suggested. We will thus consider only schemes that reduce the public key size, *i.e.* CyclicRainbow [22] and RainbowLRS2 [19, Section 9.2].

Table 2. Total reduction of Rainbow key pairs, in bytes, for variants of the scheme.

Instance	Parameters	Variant	$\|\mathcal{K}_{Pr}\|$	$\|\mathcal{K}_{Pr}^{\eta}\|$	$\|\mathcal{K}_{Pu}\|$	Difference
P-080	$(\mathbb{F}_{256}, 17, 13, 13)$	Classic	19546	6524	25740	−28.76%
		Cyclic			10618	−62.15%
		LRS2			9789	−63.98%
P-100	$(\mathbb{F}_{256}, 26, 16, 17)$	Classic	46131	12474	60390	−31.60%
		Cyclic			22246	−67.41%
		LRS2			20662	−68.89%
P-128	$(\mathbb{F}_{256}, 36, 21, 22)$	Classic	105006	24924	139320	−32.78%
		Cyclic			48411	−69.98%
		LRS2			45547	−71.16%

We compare the total key pair sizes $|\mathcal{K}_{Pr}| + |\mathcal{K}_{Pu}|$ when our method is used alongside Rainbow variants that reduce the public key size. Table 2 shows the quantity of field elements for sets of parameters from Petzoldt [19, Table 9.8]. We calculate $|\mathcal{K}_{Pu}|$ for the variants according to Eqs. 9.2 and 9.4 of the same work, and as per its Remark 9.1, note that $q = 16$ and $q = 31$ are not considered due to a security restriction of RainbowLRS2. We obtain positive results, with key pair size reductions of up to factors of 3 and no security harm to the resulting scheme.

The use of CyclicRainbow or RainbowLRS2 with the Rainbow-η_2 method is recommended. These variants are based on the linear relations described in Subsect. 3.1, and resulting implementations may be effortlessly modified to use our proposal. Moreover, in the case that higher parameters are needed, *e.g.* a security level of 256 bits, we note that the key pair will be reduced more aggressively. Thus, our results reflect changes over a wide variety of platforms and possible Rainbow deployments that benefit from lower storage requirements.

We also briefly discuss the effect of these changes on the signature generation step overall performance. In the case of Rainbow-η_1, it does not vary greatly due to the fast regeneration of the central map elements from a given PRNG and $\mathbf{S}$. On the other hand, Rainbow-η_2 uses elaborate techniques to reconstruct the central map if vinegar variables are not suitable. This process is not without cost, and it may negatively affect the average signature generation time. Still, by making use of Rainbow-η_3, these computations are entirely avoided by choosing a new salt instead of new vinegar variables, reducing the inherent overhead.

5 Conclusion

Throughout this work, we have proposed general methods to lower private key sizes that can be applied to all known Rainbow variants. We suggest fixing the first sequence of vinegar variables and reuse it on the creation of signatures, reducing the static central map storage requirements, and thus obtaining a smaller private key. Our security analysis shows that this modification creates orderly signatures and does not harm the target scheme. Furthermore, we have also addressed the problem in which no scheme could reduce both keys in the key pair, by applying our proposal to known variants that reduce the public key size. We obtain gains of up to 85% on the private key size and 71% on the total key pair size.

We propose some topics to extend this work. Evidently, it is crucial for the security of our proposal that multiple signatures do not leak information for the chosen vinegar variables. Thus, we point out that further security analysis on multi-target and side-channel attacks is desirable. We also observe that our methods directly affect the signature generation performance, since the first layer computations are moved to the key generation step. As such, we suggest that measurements are made considering the average time for signature generation, in the case that the private key has to be recomputed due to a new choice of vinegar variables.

Acknowledgements. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Additionally, we thank the anonymous referees for their suggestions.

References

1. Alagic, G., et al.: Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. Internal Report 8240, National Institute of Standards and Technology (NIST), January 2019. https://doi.org/10.6028/NIST.IR.8240
2. Bernstein, D.J., Buchmann, J., Dahmen, E.: Post Quantum Cryptography, 1st edn. Springer, Heidelberg (2008)
3. Bettale, L., Faugére, J.C., Perret, L.: Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In: Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation, pp. 67–74, July 2012. https://doi.org/10.1145/2442829.2442843
4. Billet, O., Gilbert, H.: Cryptanalysis of Rainbow. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 336–347. Springer, Heidelberg (2006). https://doi.org/10.1007/11832072_23
5. Czypek, W.: Implementing Multivariate Quadratic Public Key Signature Schemes on Embedded Devices. Master's thesis, Ruhr-Universität Bochum, April 2012
6. Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y.: Rainbow - Algorithm Specification and Documentation. Round 1 Submission, NIST Post-Quantum Cryptography Standardisation Process, December 2017
7. Ding, J., Gower, J., Schmidt, D.: Multivariate Public Key Cryptosystems, 1st edn. Springer, Boston (2006). https://doi.org/10.1007/978-0-387-36946-4
8. Ding, J., Petzoldt, A.: Current state of multivariate cryptography. IEEE Secur. Priv. **15**(4), 28–36 (2017). https://doi.org/10.1109/MSP.2017.3151328
9. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005). https://doi.org/10.1007/11496137_12

10. Ding, J., Schmidt, D., Yin, Z.: Cryptanalysis of the new TTS scheme in CHES 2004. Int. J. Inf. Secur. **5**(4), 231–240 (2006). https://doi.org/10.1007/s10207-006-0003-9
11. Ding, J., Yang, B.-Y., Chen, C.-H.O., Chen, M.-S., Cheng, C.-M.: New differential-algebraic attacks and reparametrization of rainbow. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 242–257. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_15
12. von zur Gathen, J.: CryptoSchool, 1st edn. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48425-8
13. Goldreich, O.: Foundations of Cryptography: Volume 2, Basic Applications, 1st edn. Cambridge University Press (2004)
14. Hashimoto, Y.: Cryptanalysis of the quaternion rainbow. In: Sakiyama, K., Terada, M. (eds.) IWSEC 2013. LNCS, vol. 8231, pp. 244–257. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41383-4_16
15. Hashimoto, Y.: On the security of Circulant UOV/Rainbow. Cryptology ePrint Archive, Report 2018/847, October 2018. https://eprint.iacr.org/2018/947
16. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_15
17. Kipnis, A., Shamir, A.: Cryptanalysis of the oil and vinegar signature scheme. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 257–266. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055733
18. Peng, Z., Tang, S.: Circulant rainbow: a new rainbow variant with shorter private key and faster signature generation. IEEE Access **5**, 11877–11886 (2017). https://doi.org/10.1109/ACCESS.2017.2717279
19. Petzoldt, A.: Selecting and Reducing Key Sizes for Multivariate Cryptography. Ph.D. thesis, Technische Universität Darmstadt, July 2013
20. Petzoldt, A., Bulygin, S.: Linear recurring sequences for the UOV key generation revisited. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 441–455. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37682-5_31
21. Petzoldt, A., Bulygin, S., Buchmann, J.: A multivariate signature scheme with a partially cyclic public key. In: Faugëre, J.C., Cid, C. (eds.) International Conference on Symbolic Computation and Cryptography, pp. 229–235, June 2010
22. Petzoldt, A., Bulygin, S., Buchmann, J.: CyclicRainbow – a multivariate signature scheme with a partially cyclic public key. In: Gong, G., Gupta, K.C. (eds.) INDOCRYPT 2010. LNCS, vol. 6498, pp. 33–48. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17401-8_4
23. Petzoldt, A., Bulygin, S., Buchmann, J.: Selecting parameters for the rainbow signature scheme. In: Sendrier, N. (ed.) PQCrypto 2010. LNCS, vol. 6061, pp. 218–240. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12929-2_16
24. Petzoldt, A., Bulygin, S., Buchmann, J.: Linear recurring sequences for the UOV key generation. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 335–350. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19379-8_21
25. Shim, K.-A., Park, C.-M., Baek, Y.-J.: Lite-Rainbow: lightweight signature schemes based on multivariate quadratic equations and their secure implementations. In: Biryukov, A., Goyal, V. (eds.) INDOCRYPT 2015. LNCS, vol. 9462, pp. 45–63. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26617-6_3

26. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997). https://doi.org/10.1137/S0097539795293172
27. Tang, S., Yi, H., Ding, J., Chen, H., Chen, G.: High-speed hardware implementation of rainbow signature on FPGAs. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 228–243. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_15
28. Thomae, E., Wolf, C.: Cryptanalysis of enhanced TTS, STS and all its variants, or: why cross-terms are important. In: Mitrokotsa, A., Vaudenay, S. (eds.) AFRICACRYPT 2012. LNCS, vol. 7374, pp. 188–202. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31410-0_12
29. Wolf, C., Preneel, B.: Taxonomy of Public Key Schemes based on the problem of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations. Cryptology ePrint Archive, Report 2005/077, March 2005. https://eprint.iacr.org/2005/077
30. Yasuda, T., Ding, J., Takagi, T., Sakurai, K.: A variant of rainbow with shorter secret key and faster signature generation. In: Chen, K., Xie, Q., Qiu, W., Xu, S., Zhao, Y. (eds.) ACM Workshop on Asia Public-Key Cryptography, pp. 57–62, May 2013. https://doi.org/10.1145/2484389.2484401
31. Yasuda, T., Sakurai, K., Takagi, T.: Reducing the key size of rainbow using non-commutative rings. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 68–83. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27954-6_5
32. Yasuda, T., Takagi, T., Sakurai, K.: Efficient variant of Rainbow using sparse secret keys. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. **5**(3), 3–13 (2014)
33. Yasuda, T., Takagi, T., Sakurai, K.: Efficient variant of rainbow without triangular matrix representation. In: Mahendra, M.S., Neuhold, E.J., Tjoa, M.A., You, I. (eds.) Information and Communication Technology. LNCS, vol. 8407, pp. 532–541. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-55032-4_55
34. Yi, H., Tang, S.: Very small FPGA processor for multivariate signatures. Comput. J. **59**(7), 1091–1101 (2016). https://doi.org/10.1093/comjnl/bxw008