NIST Post-Quantum Standardization Project Round 1 Multivariate Digital Signature Candidates Survey

Daniel Smith-Tone[1]

[1]National Institute of Standards and Technology,
Gaithersburg, Maryland, USA

daniel.smith@nist.gov

Abstract. This document provides a description of the multivariate digital signature schemes that have been submitted as Round 1 candidates in NIST's post-quantum standardization project. Contained in this document are a description of the field of multivariate cryptography, an overview of the history of the subfields offering context to the submissions, a description of the main cryptanalytic methods, a mathematical description of the candidate algorithms, relevant protocols and transformations, the security analyses provided in the submission supporting documentation, a performance comparison citing relevant platform issues, a review of recent comments on the schemes, and a discussion of the schemes vis-à-vis selection criteria.

Key words: Multivariate Cryptography, HFE, PFLASH, Discrete Differential, MinRank

1 Introduction

Multivariate cryptography refers to the subfield of information security constructing and analyzing primitives that are expressed and evaluated as functions of many variables. There are many variations on the theme of multivariate cryptography we ignore for our purposes, some of which intersect with other families of post-quantum cryptography, see, for example, the multivariate emulation of LWE in [1], and some of which are not central to the study or relevant to understanding the context of the Round 1 multivariate post-quantum digital signature candidates, see, for example, the schemes based on noncommutative Ore polynomials in [2]. Thus we begin with a brief overview of the main branch of multivariate public key cryptography utilizing public systems of multivariate polynomials in a commutative polynomial ring. As a note, nominally the main relevant branch of mathematics for comprehension of this area is algebraic geometry, however, in practice, rarely are any results beyond commutative algebra required.

One of the hard computational problems on which the security of many multivariate cryptosystems is based is the problem of solving systems of multivariate equations.

Problem 1 ($\mathcal{MQ}$) *The multivariate quadratic ($\mathcal{MQ}$) problem is the problem of solving a multivariate quadratic system of equations over a field. The decisional problem is known to be NP-hard over any field, and over finite fields it is NP-complete.*

Proof. Let $\mathbb{K}$ be a field and let P be an instance of SAT on n predicates, $X_1, \ldots, X_n$, where, clearly, n is at most linear in $\|P\|$. Then $Q = \{\hat{P}, x_1^2 - x_1, \ldots, x_n^2 - x_n\}$, where $\hat{P}$ represents the boolean formula in the variables x_i corresponding to P, is an $\mathcal{MQ}$ instance constructible in linear time and of length linear in $\|P\|$. Since $\mathbb{K}$ is a field, any solution must assign x_i the value 0 or 1, which is interpreted as assigning the corresponding truth value for the predicate X_i. Thus a solution to the decisional $\mathcal{MQ}$ instance Q decides P. Furthermore, if $\mathbb{K}$ is finite, then it is easy to show that the length of the calculation for any witness is polynomial in the size of the $\mathcal{MQ}$ instance.

Empirically, we have evidence that solving systems of multivariate quadratic equations is *generically* hard, so a valid technique for constructing a cryptosystem is to find a class of quadratic vector-valued functions on a vector space that is easy to invert, and transform it into a system that appears random.

Both of these tasks present challenges. The standard technique for the second task is computing a morphism of the system in an attempt to remove the properties allowing the system to be inverted. Techniques for the prior task are more varied; we discuss two methodologies for constructing efficiently invertible systems in the following subsections. As a side note, the use of these morphisms introduces some additional complexity theoretic dependencies for most multivariate schemes; we discuss these security issues and some known results in Section 3.

2 Classifications of Multivariate Cryptosystems

The field of multivariate cryptography is fairly broad and motivated by disparate ideas. In this section we offer an atlas that will be useful in comparing specific schemes. Some properties of the cryptosystems allow us to group them into families with similar properties in terms of efficiency, attack methodology and known security properties. A useful characterization of a multivariate scheme can be achieved by answering, in order: From what structure are the nonlinear multiplication operations derived? How is efficient inversion accomplished?

2.1 The Big Field Setting

One family of multivariate cryptosystems are collectively known as "big-field" schemes; though the same idea generalizes in a natural way to what one might call "big-algebra" schemes. Such schemes use the multiplicative structure of an algebra $\mathbb{A}$ over a "base" field $\mathbb{F}$, to construct an easily invertible map. The vector space structure of $\mathbb{A}$ easily allows one to pass between univariate functions over $\mathbb{A}$ and multivariate functions over $\mathbb{F}^n$. See Figure 1.

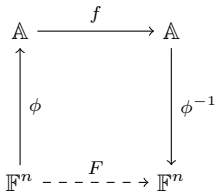


Fig. 1. The structure of a "big-algebra" map. The map ϕ is a $\mathbb{F}$-vector space isomorphism, F is a vector-valued function on $\mathbb{F}^n$, and f is an univariate function over $\mathbb{A}$.

The progenitor of all "big field" schemes is commonly known as C^*, or the Matsumoto-Imai scheme, see [3]. This scheme exploits the fact that the extension field $\mathbb{K}$ is an $\mathbb{F}$-algebra to produce two versions of a function— a vector-valued version which is quadratic over the base field, and a monomial function whose input and output lie in the extension field. Specifically, the C^* central map is the univariate function $f : \mathbb{K} \to \mathbb{K}$ defined by

$$f(X) = X^{q^\theta+1},$$

where $|\mathbb{F}| = q$, $|\mathbb{K} : \mathbb{F}| = n$, and $(q^\theta + 1, q^n - 1) = 1$. The final condition ensures that the power map is invertible in $\mathbb{K}^*$. To complete the construction, one composes invertible affine maps to produce the public key $\mathcal{P}(\mathbf{x}) = T \circ F \circ U$. The C^* scheme can be considered a sort of multivariate version of RSA; in fact, inversion of F is accomplished in exactly the same way as RSA, that is, by exponentiation by the multiplicative inverse of the encryption exponent modulo the size of the unit group.

Many schemes can be derived from this framework by altering various components of this basic structure. There are three categories of such alterations: one can choose a different central map; one can modify the central map in some specific generic way preserving efficient invertibility; or one can make one or both affine transformations non-invertible. Of course, these modifications can be taken together as well.

The cryptanalysis of the C^* scheme by Patarin in [4] inspired many big field constructions. In [5], Patarin introduced the Hidden Field Equations (HFE) cryptosystem, a natural generalization of the monomial based C^* in which the monomial map is replaced with a low degree polynomial which allows efficient inversion via the Berlekamp algorithm, see [6]. Also described by Patarin in [7] is the minus modifier (the removal of public equations) which can be applied to both HFE, producing HFE$^-$, and to C^*, creating C^{*-}, as well as the plus modifier (the addition of random equations in the central map that can be ignored on inversion) and the projection modifier (the assignment of one or more input variables to constant values before the publication of the key). In [8], the vinegar modifier (the addition of variables in the central map the values of which can be randomly assigned upon inversion) is introduced in the QUARTZ

scheme. Also, in [9], the internal perturbation modifier (the addition of a random summand with a small support) is used to produce the PMI cryptosystem.

The part of the history of "big-field" multivariate cryptography in the early 2000s in which the community was just beginning to understand the modifiers and their interactions was very tumultuous, as it was for "small-field" schemes as well. It took nearly a decade for the community to get a solid understanding of what cryptanalytic techniques were thwarted by various combinations of modifiers and to understand what weaknesses were accessible in spite of the modifiers. Eventually, in the early 2010s we developed techniques to prove security against classes of attacks that had ravaged the field earlier, see, for example [10–14] which develop the framework for proving security against differential techniques.

Though the confusion of the early 2000s illustrates that it is a good idea to remain cautious with multivariate schemes, there are a few "big-field" schemes that have survived in essentially their original form from this period or even before. The original HFE$^-$ Challenge from [5] at the 80-bit level has only recently been conclusively shown in [15] to provide no more than 79 bits of security! (The authors argue that this number may be reduced further since experiments show that the linear algebra steps seem significantly faster than the Strassen Algorithm as toy instances grow in size; still, a very small parameter change secures the scheme.) The original QUARTZ proposal in [8] uses parameters that are definitely overkill and are still secure. PFLASH originally proposed in [7] is still secure and is provably resistant to the attacks that broke its closest relatives, see [16]. Thus the technique seems viable and long-lived with some schemes over two decades old.

2.2 The Small Field Setting

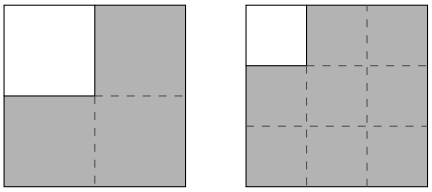
"Small field" schemes are multivariate cryptosystems that make use of only one field for generating a nonlinear map. Lacking the additional algebraic structure big field schemes use to hide easily invertible multiplication operations, small field schemes typically use a secret structure based on rank or on a partition of the variables to allow efficient invertibility. Of these two methods, one has proven to be far superior.

The first "small field" scheme was the oil-vinegar scheme of [17]. This scheme specifies two types of variables: the oil variables, which occur only linearly in the secret central map; and the vinegar variables, which occur quadratically. Thus the hidden map of the oil vinegar scheme has the form

$$\sum_{0\leq i<2n, n\leq j<2n} \alpha_{i,j,\ell} x_i x_j + \sum_{0\leq i<2n} \beta_{i,\ell} x_i + \gamma_\ell \text{ for } 0 \leq \ell < n;$$

see Figure 2(a) for a visualization of each such map. Such a function is easy to invert because assigning random values to all of the vinegar variables transforms the function into an affine map, which is easily invertible.

Clearly, this class of multivariate system is invariant under left composition by affine maps; that is to say, for all such maps F with the property that



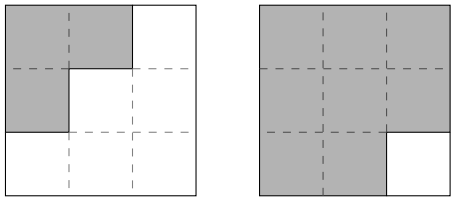
(a) OV (b) UOV

Fig. 2. The shape of the matrix representations of the polar form (or discrete differential) of each central map of (a) oil-vinegar and (b) unbalanced oil-vinegar. The shaded regions represent possibly nonzero values while unshaded areas have coefficients of zero. Note that this diagram only provides information about the quadratic terms in the central maps, as the discrete differential of the affine summand is zero.

$F(x_0, \ldots, x_{n-1}, c_n, \ldots, c_{2n-1})$ is affine for any constant $(c_n, \ldots, c_{2n-1})$ and for all affine maps T, the composition $T \circ F(x_0, \ldots, x_{n-1}, d_n, \ldots, d_{2n-1})$ is also affine for any constant $(d_n, \ldots, d_{2n-1})$. Thus, it is not necessary to compose the secret oil-vinegar map with an affine transformation mixing the outputs. The oil-vinegar scheme is then presented as $\mathcal{P}(\mathbf{x}) = F \circ L$, for some affine map L. (The name comes from the separation of oil and vinegar after they are mixed. This name does not seem to inspire much confidence since oil and vinegar separate on their own without intervention. Is it not a more intimidating task to separate salt and water or vinegar and sodium bicarbonate after mixed?)

The original proposition of oil-vinegar used the same number of oil variables as vinegar variables and was quickly broken by Kipnis and Shamir in [18]. We will provide more details of this attack in the next section. Importantly, the attack used in a critical way the balance between oil variables and vinegar variables. When the number of oil variables and vinegar variables are sufficiently separated, the attack becomes infeasible. The resulting scheme, with usually two to three times as many vinegar variables as oil variables, is called unbalanced oil-vinegar (UOV), see Figure 2(a) for a comparison between oil-vinegar and UOV. The number of equations in the system needs to be as large as the number of oil variables to make inversion likely, and having these numbers be equal is optimal.

Another scheme derived from this idea of partitioning the variables is Rainbow. Rainbow was introduced in 2005, see [19], as a generalization of UOV to many layers. Instead of every secret polynomial having the same structure, Rainbow divides the secret polynomials into layers or bands each of which have an UOV structure, but with different sets of oil and vinegar variables. Specifically, a Rainbow scheme with u layers is defined by defining the integers $0 < v_1 < \ldots < v_u < n$ and the index sets $V_1 = \{1, \ldots, v_1\}$, $V_2 = \{1, \ldots, v_2\}$, $\ldots$, $V_u = \{1, \ldots, v_u\}$. Further, we define the index sets $O_i = S_{i+1} \backslash S_i$ for $1 \leq i \leq u-1$



(a) Layer 1 (b) Layer 2

Fig. 3. The shape of the matrix representations of the polar form (or discrete differential) of each layer of the central map of a bi-layer Rainbow: (a) Layer 1 and (b) Layer 2. The shaded regions represent possibly nonzero values while unshaded areas have coefficients of zero. Note that this diagram only provides information about the quadratic terms in the central maps, as the discrete differential of the affine summand is zero.

and $O_u = \{v_u + 1, \ldots, n\}$. Then an ℓth layer Rainbow map has the form

$$\sum_{i \in O_\ell, j \in S_\ell} \alpha_{i,j,\ell} x_i x_j + \sum_{i,j \in S_\ell} \beta_{i,j,\ell} x_i x_j + \sum_{i \in S_\ell \cup O_\ell} \gamma_{i,\ell} + \delta_\ell.$$

The central map consists of $|O_\ell|$ such maps for each layer ℓ, see Figure 3 for a visualization. This allows inversion to be performed layer by layer, since assigning values to the variables indexed in V_1 transforms the layer 1 maps into affine maps in the variables indexed by O_1. After we solve for these values, we have obtained values for all of the variables indexed in V_2, and we may continually solve, layer by layer, until the values of all of the variables are recovered.

Another subfamily of "small field" multivariate schemes can be called the "step-wise triangular schemes." There are various flavors of such schemes as well as various motivations for them.

The oldest of these, known as the Tame Transformation Method (TTM) defined by T. T. Moh (also TTM!), see [20], is motivated by an important class of automorphisms of affine spaces arising in some investigations of the famous Jacobian conjecture, see [21], of algebraic geometry. This scheme was broken in [22]. The attack was denied by Moh who published a sequence of articles with tweaks or clarifications that were al broken by the same sort of idea. First, T. Moh published [23] simply denying that the attack worked while simultaneously producing a patch to prevent the attack. This paper was followed by another parameterization given in [24] that reiterated that the attack of [22] should not be trusted. Then all of these instances were broken in the paper [25].

At this point, T. Moh seems to have accepted that the attacks were real and offered a further patch in [26] which claimed to resist the above attacks with the addition of what they called "lock polynomials." Still, the new parameters were broken in [27] with an extension of the ideas developed in the previous attacks

to produce a message recovery attack bootstrapping a precomputation relying only on the public key.

Once again, T. Moh claimed to be a victim and denies the attack in [28], saying that he believed that the authors of [27] had actually used a component of the private key in the attack. He then published [29] with two new challenge instances of TTM without describing the decryption process and thus leaving secret the structure of the lock polynomials. Finally, the same team of cryptanalysts immediately broke these instances in [30] without ever learning the structure of the lock polynomials. Finally, this ugly chapter seemed to close as there was less interest in breaking the same idea multiple times with the same code.

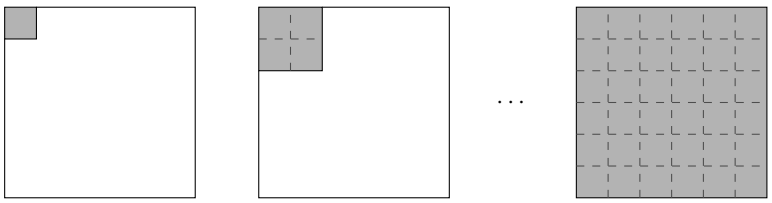
A close relative to the TTM family is the main line of triangular schemes originating in Shamir's birational permutation scheme over large rings in [31]. A very similar idea emerged which was called the sequential solution method (SSM) in [32]. These ideas were extended to construct the RSE system of [33] and was further adapted in [22] where the authors made it clear that TTM was a particular example of a more general idea that was broken. They called this more general scheme triangle-plus-minus (TPM), which was further generalized into what we now call step-wise triangular schemes (STS) in [34]. There have since been numerous variations on the theme including [35–37]. They are all very similar and the simplest exposition to provide a good understanding of all of them is to present the generic STS constructions of [34].

Unlike the oil-vinegar type schemes, the STS-style schemes require affine maps mixing both the inputs and outputs of the secret central map F. Thus a public key looks like $\mathcal{P} = T \circ F \circ U$. All such schemes can thus be modified by projection and minus modifiers to achieve different properties. The critical structure in the STS family is the structure of the central map.

The central map of a generic STS instance is defined by selecting integers $0 = u_0 < u_1 < \ldots < u_k = n$, and random quadratic maps $\mathbf{y}_i = \psi_i(\mathbf{x}_i)$, where $\mathbf{x}_i = (x_1, \ldots, x_{u_i})$ and $dim(\mathbf{y}_i) = u_i - u_{i-1}$ for $i = \{1, \ldots, k\}$. The central map is then the direct sum $\bigoplus_{i=1}^{k} \psi_i$, see Figure 4 for a visualization.

3 Overview of Known Cryptanalytic Methods and Security Constructs

There are several approaches to cryptanalysis for multivariate schemes. They can informally be characterized as algebraic, differential, rank, statistical search or quantum. This informal classification is not exclusive and is at best only loosely descriptive. For example, there are various methods for rank attacks and some of them rely on something we could call a statistical search. Another case of overlap would be bootstrapping algebraic attacks for statistical techniques. Each of these families of attacks can be employed for both "big field" and "small field" schemes, so we organize the exposition primarily by the attack type and variations and secondarily by chronological order.



(a) Layer 1 (b) Layer 2 (c) Layer k

Fig. 4. The shape of the matrix representations of the polar form (or discrete differential) of each layer of the central map of a generic STS system. The shaded regions represent possibly nonzero values while unshaded areas have coefficients of zero. Note that this diagram only provides information about the quadratic terms in the central maps, as the discrete differential of the affine summand is zero.

While this section is intended to provide a detailed overview of multivariate cryptanalysis, it will not be exhaustive. The section should elucidate, however, all of the main themes on which attacks in the literature are based. Thus, this cyclopedia should approximate fairly closely the current state of knowledge in cryptanalytic techniques.

3.1 Algebraic Attacks

Algebraic attacks directly try to solve some complexity theoretic problems involving multivariate polynomial systems. There are two principal problems and their variations that have been studied and used for cryptanalysis and security analysis in multivariate cryptography. The first of these problems is the $\mathcal{MQ}$ problem and the second is the following:

Problem 2 ($\mathcal{MP}$) *The morphism of polynomials ($\mathcal{MP}$) problem is the problem of finding a morphism of polynomials between two polynomial systems F and G; that is, of finding two affine transformations T and U such that $G = T \circ F \circ U$. The decisional problem is known to be NP-hard over any field, and over finite fields it is NP-complete, see [38].*

The main technique for directly solving an instance of $\mathcal{MQ}$ seems to always have been specifying an elimination monomial ordering refining total degree and computing a Gröbner basis for the ideal generated by the multivariate system. Such a Gröbner basis generically has polynomials with only a single variable which can then be solved efficiently reducing the dimension of the system. Thus the dominant term in the complexity of any algorithm solving $\mathcal{MQ}$ is basically the Gröbner basis computation. (As a side note, there was a great deal of work in the early 2000s on the XL family of algorithms inspired by relinearization in [39] and popularized by [40]. These algorithms are competitive with, but usually

outperformed by, modern Gröbner basis algorithms. In fact, the performance of such algorithms can be emulated by modern Gröbner basis algorithms by replacing certain function calls. Thus we do not address directly the development of XL variants.)

The first effective algorithm for computing Gröbner bases was Buchberger's algorithm presented in [41]. The algorithm is doubly-exponential in the worst case; there are known pathological examples that meet this theoretical bound. Since the late 90s, however, far superior algorithms have been developed. The best generic performance has been achieved by Faugere's F4 algorithm, see [42]. A more advanced but specialized version known as F5 can be even faster, see [43], but the exposition is very unclear and has errors. It is known that the algorithm has been successful at solving a multivariate system that was beyond the reach of F4, see [44], but it is not widely believed that the application of F5 can be as universal as was suggested in [43].

One of the first cryptanalyses by direct algebraic attack via Gröbner bases was the cryptanalysis of HFE over $GF(2)$ in [44]. This method of attack is surprisingly effective over $GF(2)$ and is often the attack which determines the parameters of multivariate schemes. This fact is interesting to note because this direct algebraic attack is usually the only known message-recovery attack for a multivariate scheme.

One particular quantity that is relevant in determining the complexity of Gröbner basis techniques is the degree of regularity. This quantity has been defined many times in an inconsistent manner. Part of the reason for this is that there are many algebraic notions of regularity. A few relevant examples are Castelnuovo-Mumford regularity (the minimum difference of the index of a free module and the largest degree among its generators in a minimal graded free resolution of the ideal as a graded module over the polynomial ring), Hilbert regularity (the dimension beyond which the Hilbert function of the quotient of the polynomial ring by the ideal becomes a polynomial), the solving degree (the degree at which linear algebra alone produces minimal generators forming a Gröbner basis), the semi-regular degree (the smallest degree of a nontrivial relation between members of the ideal assuming that the ideal has as few relations as is possible) and the first fall degreee (the degree at which a nontrivial syzygy, that is, a symmetry that is not generic to all ideals regardless of dimension, is found). Of these, evidence seems to indicate that Castelnuovo-Mumford regularity and the solving degree may almost always be equal, they are usually bounded from above by the Hilbert regularity, the smallest of these is the first fall degreee, and the semi-regular degree is somewhere between the first fall degreee and the Hilbert regularity. Most careful analysis of multivariate systems tries to measure the first-fall degreee and conservatively assumes that the solving degree is essentially the same. This assumption can definitely fail for multivariate systems in general and makes a meaningful difference in the rank analysis of LRPC schemes.

The best known method for directly solving an instance of $\mathcal{MQ}$ (at least asymptotically over fields larger than $GF(2)$) is known as the hybrid technique.

Let F be a multivariate system of m equations in n indeterminants. A basic result from commutative algebra is that the expected dimension of the space of solutions is $n - m$. If $m > n$, then we say that the system is overdetermined, if $m = n$ the system is determined, and if $m < n$ the system is said to be underdetermined. For any underdetermined system it is more efficient to preface a Gröbner basis calculation with a random assignment of $m - n$ variables to constant values. It is still expected that there is a zero-dimensional solution space, which can be found via the Gröbner basis calculation. In some instances, however, and in particular if the system is defined over a small field, it can be more efficient to guess some additional variables before computing the Gröbner basis. Since the first fall degreee is monotonically nonincreasing under proper containment of generators, adding additional relations by setting more variables can decrease the complexity of the Gröbner basis step. The hybrid technique attempts to optimize the tradeoff between guessing correct values for variables and decreasing the first fall degreee.

A significant issue of the main Gröbner basis techniques, such as F4, is that *time* is on the order of s^ω where $2 \leq \omega < 3$ is the linear algebra constant and s is the size of some large matrix in the computation, while *space* is on the order of s^2. We are typically justified in saying that ω is no larger than 2.81 as long as we can use a few bits to handle the constants in the big-oh, but an attack at, say, the 256-bit level must still use space on the order of 2^{183}. This means that if we can barely break a scheme with F4 at the 256-bit security level, then assuming that we can create atomic scale memory with an average atomic weight comparable to carbon, it would require more than 300 Earth-masses of memory to complete the calculation.

As a final note on the $\mathcal{MQ}$ problem, there is a body of work devoted to the $\mathcal{MQ}$ problem over $GF(2)$. This line of research is directed at optimizing exhaustive search methods as well as alternative algorithms in the case of $GF(2)$. A particular requirement of these algorithms is efficiency in memory usage, addressing the above concern about the feasibility of cryptanalysis via F4. Some results in this direction include [45], analyzing exhaustive search, [46], an interesting new advance in solving boolean systems that appears to be the current record-holder for classical CPU system solvers, and [47], applying the above techniques in GPU arrays.

The $\mathcal{MP}$ problem actually lends itself to a very similar analysis. The main method of directly solving the $\mathcal{MP}$ problem in the quadratic case is to model it as a system of bi-affine equations and try to solve the system directly via Gröbner basis techniques. There are techniques that marginally speed up Gröbner basis calculations for bi-affine systems, see [48]; however, the instances of $\mathcal{MP}$ relavant for cryptography are very large and this technique does not seem to be useful.

There are some other variations on the $\mathcal{MP}$ problem that are relevant to cryptography. One is the Isomorphism of Polynomials (IP) problem. This problem addresses instances of $\mathcal{MP}$ where it is known that the affine maps must be invertible. It was shown in [38] that this problem is not NP-hard unless the polynomial-time hierarchy collapses to the second level. An additional related

problem is the IP1S problems (refering to IP with one secret affine transformation). This problem is claimed to be in P in the quadratic case with a solution given in [49]. The last the author checked, there is an error in Proposition 9, therein, though it seems that Proposition 9 might be true generically. Thus, the result is probably both wrong and trustworthy. A last important note is due: the *statement* of the IP1S instance provides two potentially isomorphic systems of polynomials; thus, the solution to IP1S only breaks a scheme like UOV if one can successfully guess the nonlinear portion of the private key. Therefore, the significance of results like [49] for schemes with an appropriate amount of entropy in the private key space is primarily in providing tools for accurately counting equivalent keys, thus establishing the direct connection between entropy in the private key space and entropy in the public key space.

3.2 Differential Attacks

A second class of attacks, the differential attacks, utilize a special form of differential that can be useful in translating certain properties of nonlinear functions into properties of linear maps that can be analyzed with linear algebra techniques. The differential of a field map, f, is defined by

$$Df(a,x) = f(a+x) - f(a) - f(x) + f(0).$$

Various cryptanalyses can be viewed through the lense of differential techniques. It is reasonable to say that a general lack of understanding of differential security caused many of the growing pains in "big field" multivariate cryptography through the mid 2000s.

As an example, we can even consider Patarin's initial attack, in [4], on Imai and Matsumoto's C^* scheme, see [3], as the exploitation of a trivial differential symmetry. Suppose $f(x) = x^{q^\theta+1}$ and let $y = f(x)$. Since the differential of f, Df, is a symmetric bilinear function,

$$\begin{aligned} 0 = Df(y,y) &= Df(y, x^{q^\theta+1}) \\ &= yx^{q^{2\theta}+q^\theta} + y^{q^\theta}x^{q^\theta+1} \\ &= x^{q^\theta}(yx^{q^{2\theta}} + y^{q^\theta}x). \end{aligned}$$

Dividing by x^{q^θ} we have Patarin's linear relation, $yx^{q^{2\theta}} = y^{q^\theta}x$; see [50] for details.

Differential methods provide powerful tools for decomposing a multivariate scheme. To illustrate the versatile nature of differential attacks, we review the attack of Kipnis and Shamir, see [18], on a non-big-field system, the oil and vinegar scheme. Though they use differing terminology, the attack exploits a symmetry hidden in the differential structure of the scheme.

Recall that the oil and vinegar scheme is based on a hidden quadratic system of equations, $f : k^n \to k^o$, in two types of variables, $x_1, ..., x_o$, the oil variables, and $x_{o+1}, ..., x_{o+v=n}$, the vinegar variables. We focus on the balanced oil and

vinegar scheme, in which $o = v$. Let $c_1, ..., c_v$ be random constants. The map f has the property that $f(x_1, ..., x_v, c_1, ..., c_v)$ is affine in $x_1, ..., x_v$. The encryption map, $\overline{f}$ is the composition of f with an n-dimensional invertible affine map, L.

Let O represent the subspace generated by the first v basis vectors, and let V denote the cosummand of O. Notice that the discrete differential given by $Df(a, x) = f(x + a) - f(x) - f(a) + f(0)$ has the property that for all a and x in O, $Df(a, x) = 0$. Thus for each coordinate, i, the differential coordinate form Df_i can be represented:

$$Df_i = \begin{bmatrix} 0 & Df_{i1} \\ Df_{i1}^T & Df_{i2} \end{bmatrix}.$$

Let M_1 and M_2 be two invertible matrices in the span of the Df_i. Then $M_1^{-1}M_2$ is an O-invariant transformation of the form:

$$\begin{bmatrix} A & B \\ 0 & C \end{bmatrix}.$$

Now the Df_i are not known, but $D(f \circ L)_i = L^\top Df_i L$, so the $L^\top Df_i L$ are known. Notice that if M is in the span of the Df_i, then $L^\top ML$ is in the span of the $L^\top Df_i L$. Also, since $(L^\top M_1 L)^{-1}(L^\top M_2 L) = L^{-1}M_1^{-1}M_2 L$, there is a large space of matrices leaving $L^{-1}O$ invariant, which Kipnis and Shamir are able to exploit to effect an attack against the balanced oil and vinegar scheme; see [18] for details. Making the oil and vinegar scheme unbalanced, see [51], corrects this problem by making it very unlikely that any subspace is invariant under a general product $M_1^{-1}M_2$.

While the first of the above attacks exploits a symmetric relation exhibited by the differential of the central map, the second attack utilizes an invariant of the differential. Both of these ideas generalize and have been used as the basis for cryptanalysis (as opposed to these examples which are fit into the differential cryptanalysis mold after the fact).

The first cryptanalysis that was actually discovered using differential techniques was the attack by Dubois et al. in [52] of a popular iteration of C^{*-} called SFLASH, see [53]. SFLASH was extremely efficient and was fast even on the cheapest smart cards. The attack broke SFLASH by way of a symmetric differential relation present in the central monomial map. Note that for $f(x) = x^{q^\theta+1}$, that $Df(a, x) = ax^{q^\theta} + a^{q^\theta}x$. Therefore, for any element $\sigma \in \mathbb{K}$, we obtain the relation

$$Df(\sigma a, x) + Df(a, \sigma x) = (\sigma^{q^\theta} + \sigma)Df(a, x). \tag{1}$$

Composing with the affine transformations T and U we obtain the following relation on the C^* scheme $\mathcal{P}$ without the minus modifier

$$D\mathcal{P}(\mathbf{N}_\sigma \mathbf{a}, \mathbf{x}) + D\mathcal{P}(\mathbf{a}, \mathbf{N}_\sigma \mathbf{x}) = \mathbf{\Lambda}_\sigma D\mathcal{P}(\mathbf{a}, \mathbf{x}),$$

where $\mathbf{N}_\sigma$ is of the form $U^{-1}\mathbf{M}_\sigma U$ where $\mathbf{M}_\sigma$ is the matrix representation over $\mathbb{F}_q$ of left multiplication by σ. Since the discrete differential operator D is linear, for any linear map Π, $\Pi \circ Df = D(\Pi \circ f)$; thus, applying the minus projection

Π to the above equation we reveal the following differential symmetry on the C^{*-} public key $\mathcal{P}_\Pi$.

$$D\mathcal{P}_\Pi(\mathbf{N}_\sigma \mathbf{a}, \mathbf{x}) + D\mathcal{P}_\Pi(\mathbf{a}, \mathbf{N}_\sigma \mathbf{x}) = \mathbf{\Lambda}'_\sigma D\mathcal{P}(\mathbf{a}, \mathbf{x}).$$

Here, we notice that given a map $\mathbf{N}_\sigma$ we can produce linear combinations of all coordinates of $D\mathcal{P}$. In this way, we can recover enough linearly independent coordinates of $D\mathcal{P}$ to produce a full rank C^* public key compatible with the C^{*-} key, at which point Patarin's linearization equations attack breaks the scheme completely. (It is important to note that the above symmetry is linear in the unknown coefficients of $\mathbf{N}_\sigma$ and $\mathbf{\Lambda}'_\sigma$. Furthermore, the map $D\mathcal{P}$ on the right side is not known completely. So the technique to derive such a map $\mathbf{N}_\sigma$ is to require that the first few coordinates of the left hand side lie in the span of the known coordinates of the differential. A simple calculation shows that this technique works for the original parameters of SFLASH and with a distillation technique, it was shown that as many as half of all equations could be removed by the minus modifier and still the attack works.)

The SFLASH attack relied on the multiplicative symmetry of Equation 1 which looks like the expression in the differential of the fact that f is a multiplicative function. Still, the attack only seems to require that some $\mathbb{F}_q$-linear maps can be filtered through the differential; in particular, the same attack broke some other schemes that were variations on the theme of SFLASH, see [54], as an example, breaking the ℓIC scheme of [55]. This observation motivates a more general definition of a linear differential symmetry.

Definition 1 *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be $\mathbb{F}_q$-quadratic. Then f is said to have a linear differential symmetry if there exist affine maps L and Λ_L such that*

$$Df(L\mathbf{a}, \mathbf{x}) + Df(\mathbf{a}, L\mathbf{x}) = \Lambda_L Df(\mathbf{a}, \mathbf{x}).$$

It is easy to prove that this is the most general form of linear symmetric relation on the differential of such a vector function. This statement means that given any relation on the differential of a map f which is linear in the coefficients of linear maps, then f satisfies a relation of the above form. Thus, this object is the only thing one needs to study to fully understand differential symmetry.

The nice thing about such a generalized and abstract definition encompassing all such attacks is that it provides a tool for precisely determining when such attacks are viable. In [11], the subspace of all linear maps L inducing a linear differential symmetry is derived for C^*, SFLASH, and the SQUARE cryptosystem, see [56]. This result is bootstrapped from the results on the multiplicative symmetry first presented in [10]. It is also shown that the projected SFLASH (where both affine transformations are singular), known as PFLASH is immune to differential symmetric attacks as suggested in [57] provided that the projection map and C^* map satisfy some mild conditions. Finally, a heuristic argument is also provided in [11] that HFE has no linear differential symmetry.

In [12], a generalization of the differential invariant version, presented earlier in this section, of the Kipnis-Shamir attack on oil-vinegar of [18] is produced. The authors defined a differential invariant in the following way.

Definition 2 *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$ be a quadratic map. We say that f has a differential invariant (V, W) if $\langle \mathbf{Ax}, \mathbf{y} \rangle = \mathbf{0}$ for all $\mathbf{x} \in V$, all $\mathbf{y} \in W^\perp$ and all $\mathbf{A}$ in the span of Df, where $dim(W) \leq dim(V)$.*

The definition is designed to capture the phenomenon of all coordinates of the differential simultaneously sending some fixed subspace into another fixed subspace. (This occurence is the weird action taking place in the oil-vinegar map. The oil subspace is simultaneously mapped into the vinegar subspace by all coordinates of the differential.) The authors then proved that several "big field" schemes do not have differential invariants. The entire discussion was highly speculative since the only example provided of an actual cryptanalysis involving differential invariants was only classified as a differential invariant attack a posteriori. Thus, the paper seems to make a claim something like, "This fence will contain any unicorn."

By some manner of fortune, the differential invariant technique was soon justified (at least in a slightly generalized form) by a differential invariant attack, see [58], on the ABC Simple Matrix Encryption Scheme of [59]. This attack and the subsequent variations on it, see [60–62], generalized Definition 2 to *subspace* differential invariants, where $\mathbf{A}$ is only required to lie in some specified subspace of the span of Df.

The above attacks justify differential invariant analysis, and reveal an interesting relationship between differential invariants and certain rank defects in multivariate schemes. In particular, each of these attacks is a form of a rank attack with what is called in [63] *interlinked kernels*. The distinction between these differential invariant attacks and the older rank attacks on interlinked kernels is that the invariant structure allows the attacker to use the "invariant trick" demonstrated in the Kipnis-Shamir attack to somewhat speed up the analysis. Another important distinction is that the precise definition makes proving that a map has no differential invariants straightforward. This analysis was used to prove differential security for HFE, HFE$^-$, HFEv$^-$, ZHFE (see [64]), and PFLASH in [65, 14, 66, 67, 16].

One can consider the Rainbow Band Separation attack of [68] a differential attack on the Rainbow scheme. The attack is based on the fact that given the central map of a Rainbow instance f, there exists a linear map L projecting onto the last oil layer such that $Df(L\mathbf{a}, L\mathbf{x}) = \mathbf{0}$. One can let the unknown coefficients of this map L be variables and define a quadratic system that is overdefined. Empirically, these systems appear to be semi-regular, so we can easily estimate the degree of regularity of such systems. The complexity of such an attack is $\mathcal{O}\left(\binom{n+d_{reg}}{d_{reg}}^o mega\right)$, where *omega* is the linear algebra constant.

3.3 Rank Attacks

Another historically important class of attack is the rank attack. There are several versions of this attack and they are applicable in general for "big field" schemes and "small field" schemes. In addition to the variety of types of rank attack, there are also multiple techniques for implementing such attacks. We will

classify rank attacks into four overlapping camps: 1) low rank— recovering a low rank linear combination of the matrix representations of the public quadratic forms (or their differentials); 2) high rank— recovering a linear combination of the public quadratic forms representing a polynomial avoiding a large class of monomials; 3) dual rank— recover a small subspace that is in the kernel of every matrix in a large subspace of the span of the public quadratic forms; and 4) Q-rank— recover a linear combination of the public quadratic forms that is low rank when viewed as a quadratic form on an algebra over an extension field. The fourth category is only a reasonable possible weakness for "big field" schemes; however, all of the first three categories are applicable to both "big field" and "small field" schemes.

The first of these attacks, the low rank attack, is equivalent to a well-studied problem, the MinRank problem. The computational MinRank problem can be stated as follows. Given k matrices of dimension $p \times q$ over a field $\mathbb{F}$, find a linear combination of rank at most r. The decisional version of the MinRank problem is known to be NP-complete, see [69], and seems difficult in practice.

There are three main methods for solving the MinRank problem; the most efficient depends on the parameters of the instance. The first method for solving this problem in the cryptonomy used Kipnis-Shamir modeling with relinearization, see [39].

A little setup is necessary to get to what the Kipnis-Shamir modeling actually is. In this paper, the authors present an attack on the HFE cryptosystem exploiting the fact that the quadratic form representing the HFE central map over an algebra over the extension field has low rank. This quantity is known as the Q-rank of the map. Specifically, we consider the $\mathbb{K}$-algebra $\mathbb{A} = \{(a, a^q, a^{q^2}, \ldots, a^{q^{n-1}}) : a \in \mathbb{K}\}$. Then a map of the form

$$f(x) = \sum_{q^i+q^j<D} \alpha_{i,j} x^{q^i+q^j}$$

can be equivalently written as a quadratic form on $\mathbb{A}$:

$$\left(x\ x^q \cdots x^{q^{n-1}}\right) \begin{pmatrix} \alpha_{00} & \alpha_{0,1}/2 & \cdots & \alpha_{0,r-1}/2 & 0 \cdots 0 \\ \alpha_{0,1}/2 & \alpha_{1,1} & \cdots & \alpha_{1,r-1}/2 & 0 \cdots 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \ddots \vdots \\ \alpha_{0,r-1}/2 & \alpha_{1,r-1}/2 & \cdots & \alpha_{r-1,r-1} & 0 \cdots 0 \\ 0 & 0 & \cdots & 0 & 0 \cdots 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \ddots \vdots \\ 0 & 0 & \cdots & 0 & 0 \cdots 0 \end{pmatrix} \left(x\ x^q \cdots x^{q^{n-1}}\right)^\top,$$

where $r - 1 < log_q(D)$. Thus, we would say that the map f has Q-rank r. The method presented in [39] is to interpolate a representation of the public key as an univariate map over $\mathbb{K}$, compute all of the Frobenius powers of this map, write down their matrix representations as quadratic forms on $\mathbb{A}$, and solve the resulting MinRank instance with field $\mathbb{K}$.

To solve the MinRank instance, they suggested constructing a large system of equations in the following way. Let $\mathbf{M}_1, \ldots, \mathbf{M}_k$ represent the matrices. Compute the sum $\mathbf{X} = \sum_{i=1}^{k} x_i \mathbf{M}_i$ and construct the matrix

$$\mathbf{K} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ k_1 & k_{r+1} & \cdots & k_{(n-r-1)r+1} \\ k_2 & k_{r+2} & \cdots & k_{(n-r-1)r+2} \\ \vdots & \vdots & \ddots & \vdots \\ k_r & k_{2r} & \cdots & k_{(n-r)r} \end{pmatrix},$$

where the x_i and k_i are variables. They then proposed what they claimed to be an efficient algorithm for solving such a nonlinear system that they called relinearization. The efficiency of relinearization seems after more research to be very poor in comparison to other standard techniques, but this method of modeling the MinRank instance and its complexity are active areas of research even today.

The second published MinRank cryptanalysis technique can be called "linear algebra search" and appeared in [22]. The idea is simple. Suppose the low rank matrix in the span of the public quadratic forms $\mathbf{M}_1, \ldots, \mathbf{M}_m$ is given by the sum $\sum_{i=1}^{m} x_i \mathbf{M}_i$. Randomly select a vector $\mathbf{w}$. If we are very lucky, this vector is in the kernel of the low rank linear combination. Thus we have the matrix equation

$$\sum_{i=1}^{m} x_i \mathbf{M}_i \mathbf{w}^\top = \mathbf{0}.$$

If the dimension of $\mathbf{w} = n$ is as large as m, then we can solve for the indeterminants x_i with linear algebra. Otherwise, we need to select more than one such vector $\mathbf{w}$ at a time. This technique has the benefit of admitting a trivial complexity analysis. Since the target matrix has rank r, its kernel is of dimension $n-r$, and so the probability that $k = \lceil \frac{m}{n} \rceil$ independent vectors are in the kernel is q^{-kr}. Taking into consideration the cost of linear algebra, the complexity of this approach is $q^{kr} m^\omega$ where ω is the linear algebra constant. (This quantity can be tweaked a bit when m/n is only slightly larger than an integer.)

The third MinRank technique in cryptography literature is the minors modeling technique popularized in [70]. Once again, this method begins with creating the matrix of linear forms $\sum_{i=1}^{m} x_i \mathbf{M}_i$. The technique exploits the fact that if there is a matrix in the span of the $\mathbf{M}_i$ of rank at most r, then the ideal generated by the $r+1$-dimensional minors of this formal linear combination has a nonempty ideal over the relevant field. Thus, the MinRank instance can be attacked via Gröbner basis methods with only m variables (as opposed to the Kipnis-Shamir methodology which involves $m+(n-r)r$ variables). This method also has a credible complexity analysis based on the genericity conjecture of [71,

Conjecture 1], which is related to Fröberg's Conjecture, see [72]. For the sizes of most instances of relevance for multivariate cryptography, the complexity seems to be $\mathcal{O}\left(\binom{m+r}{r}^{\omega}\right)$, where ω is the linear algebra constant.

It should be stated that the Kipnis-Shamir modeling produces a much larger system than that of minors modeling in terms of the number of variables. In practice, though, there are many instances in which the Kipnis-Shamir approach seems faster. We are currently investigating this phenomenon and have some preliminary results indicating why this is the case. Furthermore, the Kipnis-Shamir approach lends itself better to some simple optimizations that can be easily shown to be faster than minors for some schemes. Thus an honest report must indicate that the complexity of this problem is not solidly understood, though we appear to be on the threshold of such an understanding.

The second rank attack in our list, the high rank attack, is useful in discovering formulae avoiding a large collection of monomials. This attack is almost identical to the dual rank attack (which is often also called a high rank attack) which attempts to find variables occurring in few formulae. Since the target of the dual rank attack is a special case of the target of a high rank attack, dual rank attacks are high rank attacks. A high rank attack is not necessarily a dual rank attack, however. An example of a high rank attack that is of a different form than the dual rank attacks is the cryptanalysis in [73] of the PMI scheme of [9]. A much simpler exposition of the attack and an interesting relationship between certain rank based attacks and some trivial differential invariant attacks is presented in [74]. This attack also a good example of a statistical technique, so a more careful exposition will be provided in Subsection 3.4.

The dual rank attack is actually dual to the low rank attack. Whereas the search version of the low rank attack works by finding a large kernel shared by a small subspace of the public quadratic forms, the dual attack finds a small kernel shared by a large subspace of the public forms.

The technique is essentially the reverse of the linear algebra search technique for low rank attacks. Instead of guessing a kernel variable, solving for a linear combination of matrices and checking that the rank of that combination is low, the dual rank attack guesses a linear combination, computes the kernel, and checks that the subspace of maps sharing the kernel is large. The algorithm is as follows: 1) Compute a random linear combination $\mathbf{M} = \sum_{i=1}^{m} \alpha_i \mathbf{M}_i$; 2) Find $K = ker(\mathbf{M})$; and 3) Check that the linear system $\sum_{i=1}^{m} x_i(\mathbf{M}_i K)$ is of low rank (equivalently, setting this sum equal to zero, there is a large dimensional solution space of x_is). If the scheme has a collection of variables occurring quadratically in at most d formulae, then with probability q^{-d}, K is the annihilator of the complement of that collection. Thus the complexity is something like $m^3 q^d$ for this attack. This specific method seems to have originated in [22].

Essentially all of these attacks can be used against essentially all of the stepwise triangular schemes. Which technique is most efficient depends on the parameters of the scheme. It is important to note that since the central map of Rainbow contains maps of different rank by design, the low and dual rank at-

tacks are applicable, though the design parameters are always selected to make these attacks ineffective.

3.4 Statistical Attacks

Statistical attacks are attacks that gather data by computing some quantity whose distribution is known to discover a some of the secret structure of the scheme. Many of the techniques already mentioned can be considered statistical (for example linear algebra search for MinRank), but some attacks use statistical means in a different and more intrinsic way. For example, some attacks use statistics that are less discrete than, say, matrix rank for disambiguation. Still other attacks gather data differently to infer structure from emergent properties in the data stream instead of finding a single outlier.

An example of a few statistical techniques using less discrete metrics include the rank and algebraic attacks using projection of [75]. These methods combine random projections of the public key of HFEv$^-$ with attacks based on Q-rank or direct algebraic attacks. The attacks attempt to project away from a subspace of the vinegar variables to reduce the Q-rank of the central map. When this occurs, the first fall degreee can change, which can be detected. Even more, there are instances where the time for solving the system changes or matrix size changes more than expected even at the same first fall degreee. These methods form the basis of a distinguisher using algebraic techniques. While the main focus of these attacks is in some form a type of rank attack, the way the data are handled and interpreted are fundamentally different.

An example of an attack infering structure in an emergent way from a data stream is the cryptanalysis of the PMI scheme of [9] presented in [73]. A simpler explanation is provided in [74] which we reiterate here.

The internally perturbed C^* scheme, PMI, see [9], uses the idea of adding a random summand of low dimensional support to the core map. Given the standard parameters of C^*, internal perturbation augments the central map, f, with a summand $g \circ L$, where $g : \mathbb{F}_q^l \to \mathbb{K}$ is a random quadratic map and $L : \mathbb{K} \to \mathbb{F}_q^l$ is a random $\mathbb{F}_q$-linear map. Thus the entire encryption map is given by:

$$\mathcal{P}(\mathbf{x}) = T \circ \phi^{-1} \circ f \circ \phi \circ U(\mathbf{x}) + T \circ \phi^{-1} \circ g \circ L \circ \phi \circ U(\mathbf{x}).$$

Specifically, the map $\mathbf{y} = \mathcal{P}(\mathbf{x})$ can be "inverted" by computing all possible outputs, z, of the random quadratic, g, subtracting $T \circ \phi^{-1}(z)$ from $\mathcal{P}(\mathbf{x})$, and applying the decryption routine of C^* to the result. If the output, $\mathbf{x}$, of this procedure matches a preimage of z under $g \circ L \circ U$, then $\mathcal{P}(\mathbf{x}) = \mathbf{y}$ and $\mathbf{x}$ is legitimately an inverse of $\mathbf{y}$. If none of the q^l values of z share such a preimage with the C^* portion of the map, then $\mathbf{y}$ is not in the image of $\mathcal{P}$.

With a change of basis we can express L as $\tilde{L}$ with matrix form:

$$\tilde{\mathbf{L}} = \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}.$$

We then have:

$$\mathcal{P}(\mathbf{x}) = \tilde{T} \circ \phi^{-1} \circ \tilde{f} \circ \phi \circ \tilde{U}(\mathbf{x}) + \tilde{T} \circ \phi^{-1} \circ \tilde{g} \circ \phi \circ \tilde{L} \circ \tilde{U}(\mathbf{x}),$$

and in this basis the differential of each formula in the central map has the form:

$$D\tilde{f}_i + D(\tilde{g}_i \tilde{L})_i = \begin{bmatrix} \mathbf{D}\tilde{\mathbf{f}}_{i1} & \mathbf{D}\tilde{\mathbf{f}}_{i2} \\ \mathbf{D}\tilde{\mathbf{f}}_{i2}^{\top} & \mathbf{D}\tilde{\mathbf{g}}_i + \mathbf{D}\tilde{\mathbf{f}}_{i3} \end{bmatrix}.$$

One may note that for n odd, without the g component, each differential coordinate form has corank 1. If g is truly randomly selected, then often when $L(\phi(U(\mathbf{x})))$ is nonzero, the rank of the differential coordinate form will be smaller. An equivalent observation resulted in an attack discovering the "noise kernel," effectively removing the perturbation, see [73].

3.5 Quantum Attacks and Speedups

The main known impact of quantum algorithms on multivariate cryptanalysis is the Grover quadratic speedup, see [76] in different guises. There are several attacks that perform a memoryless search such as the linear algebra search technique for solving MinRank that seem to be able to utilize the full power of the quadratic speedup. Others, such as the hybrid algebraic attacks which preface a long and memory intensive calculation with a search seem less likely to be impacted; however, we still cautiously include the quadratic speedup in our analyses. Finally, there is at least one recent advance that seems to be a legitimate technique to be considered.

This result is the recent article outlining a quantum implementation of the FXL algorithm and its analysis, see [77]. The analysis shows that the quantum FXL algorithm beats Grover search on generic systems over $\mathbb{F}_2$. The complexity is approximately $2^{0.45n}$, which has practical implications. In particular, any scheme that was agressively setting parameters to thwart Grover search must, in fact, use keys that are 11% larger.

4 NIST PQC Round 1 Candidates

In this section we present each of the multivariate signature submissions to the NIST Post-Quantum Standardization Project. In each subsection, we begin by providing the original proposal of each scheme in the literature, if it is previously published, classify the scheme relative to the characterizations presented in Section 2, describe the primitive with the parameters provided in the proposal and present any transformations used to achieve security properties.

4.1 Gui

The Gui multivariate signature scheme was first proposed at Asiacrypt 2015 in [78]. Gui is a "big-field" scheme in the HFEv$^-$ family of digital signature

schemes. The parameters for Gui have been chosen after a great many discoveries impacting the feasibility and complexity of attacking such schemes via differential, algebraic or statistical methods, see [14, 79–81, 75]. Thus, there is a giant performance improvement between the parameters of the HFEv$^-$ scheme QUARTZ, see [8], the odd characteristic parameters first provided with Gui in [78] and the final form of Gui in the submission, see [82].

Let $\mathbb{F} = \mathbb{F}_q$ be a finite field with q elements and let $\mathbb{E}$ be a degree n extension of $\mathbb{F}$. Let $\phi : \mathbb{F}^n \to \mathbb{E}$ be an $\mathbb{F}$-vector space isomorphism. Let D be the degree bound of the HFE polynomial and let $r = \lceil log_q(D) \rceil$. Let a be the corank of the minus modifier and let v be the number of vinegar variables added by the vinegar modifier. Let k be the repetition factor in the Feistel-Patarin construction, see [83]. This method of k linked inversions of the primitive avoids Patarin's attack described in his *l'habilitation à diriger des recherches*, see [84]. Let $\mathcal{S} : \mathbb{F}^n \to \mathbb{F}^{n-a}$ be a random full rank affine projection and let $\mathcal{T} : \mathbb{F}^{n+v} \to \mathbb{F}^{n+v}$ be an invertible affine transformation.

The central map of Gui is $\mathcal{F} : \mathbb{E} \times \mathbb{F}^v \to \mathbb{E}$ defined by

$$\mathcal{F}(X, \mathbf{y}) = \sum_{0 \leq i \leq j}^{q^i + q^j \leq D} \alpha_{i,j} X^{q^i + q^j} + \sum_{0 \leq i}^{q^i \leq D} \beta_i(\mathbf{y}) \cdot X^{q^i} + \gamma(\mathbf{y}),$$

where $\alpha_{i,j} \in \mathbb{E}$, $\beta_i : \mathbb{F}^v \to \mathbb{E}$ are affine and $\gamma : \mathbb{F}^v \to \mathbb{E}$ is quadratic. The public key is then given by

$$\mathcal{P} = \mathcal{S} \circ \phi^{-1} \circ \mathcal{F} \circ (\phi \otimes id_v) \circ \mathcal{T}.$$

From these formulae, we can derive the public and private key sizes. Their formulae in terms of bits are

$$(n-a)(n+1) + (n+v)(n+v+1) + n\left[\binom{r+1}{2} + r(v+1) + \binom{v+2}{2}\right],$$

and

$$(n-a)\frac{(n+v+1)(n+v+2)}{2},$$

respectively. The sizes of the keys are provided for the proposed parameter sets in Figure 5.

Signature Generation: To generate a signature on a document represented by d, one computes

$$\mathbf{h} = \mathcal{H}(d) \| \mathcal{H}^2(d) \| \cdots \| \mathcal{H}^\ell(d)$$

of sufficient length to encode k vectors $\mathbf{d}_i$ for $1 \leq i \leq k$ in $\mathbb{F}^{n-a}$. We then apply Algorithm 1

Signature Verification: To verify a signature for document d, one first computes the formated hash values $\mathbf{d}_i$ as above. Then we apply Algorithm 2.

In this form, Gui provides universal unforgeability. The submitters transform Gui into an EUF-CMA (Existential Unforgeability under Chosen Message Attack) scheme by using the transformation in [85]. Specifically, instead of signing the document d, a signer chooses a random 128-bit "salt" r, computes a

Parameter Set	q	n	r	v	a	k
Gui-184	2	184	6	16	16	2
Gui-312	2	312	8	24	20	2
Gui-448	2	448	10	32	28	2

	Private Key Size	Public Key Size	Signature Size (w/ salt)
Gui-184	19.1kB	416.3kB	360b
Gui-312	59.3kB	1955.1kB	504b
Gui-448	155.9kB	5789.2kB	664b

Fig. 5. The parameters of the Gui submissions and their private and public key sizes.

Algorithm 1 GuiSign

Input: $\mathcal{S}, \mathcal{F}, \mathcal{T}, \phi, k$ and $\mathbf{d}_i$ for $1 \leq i \leq k$.
Output: Gui signature $\sigma \in \mathbb{F}^{(n-a)+k(a+v)}$.

$\mathbf{S}_0 \leftarrow \mathbf{0}$
for all i **from** 1 **to** k **do**
  $\mathbf{s} \leftarrow \mathbf{d}_i \oplus \mathbf{S}_{i-1}$
  $\mathbf{x} \xleftarrow{\$} \{\mathbf{x} : \mathcal{S}(\mathbf{x}) = \mathbf{s}\}$
  $X \leftarrow \phi(\mathbf{x})$
  repeat
    $\mathbf{y} \xleftarrow{\$} \mathbb{F}^v$
    $U \leftarrow$ Cantor-Zassenhaus-Root$(\mathcal{F}(\cdot, \mathbf{y}) = X)$
  until C-Z-Root is unique
  $\mathbf{u} \leftarrow \phi^{-1}(U) \| \mathbf{y}$
  $\mathbf{z} \leftarrow \mathcal{T}^{-1}(\mathbf{u})$
  $\mathbf{S}_i \| \mathbf{X}_i \leftarrow \mathbf{z}$, where $\mathbf{X}_i \in \mathbb{F}^{n-a}$ and $\mathbf{X}_i \in \mathbb{F}^{a+v}$
end for
return $\sigma = \mathbf{S}_k \| \mathbf{X}_k \| \cdots \| \mathbf{X}_1$

signature in the above manner for the message $\mathcal{H}(d)\|r$, and appends r to the signature. Upon verification, r is recovered by parsing the signature, and the verifier verifies the message $\mathcal{H}(d)\|r$.

The description in [82] of the signing algorithm indicates the important step of the Cantor-Zassenhaus algorithm performed. This operation is a polynomial GCD calculation involving a polynomial of degree q^n, namely $Y^{q^n} - Y$. Since this step may be confusing for the ready, later in the document the authors explain the standard trick for this calculation, which is to compute repeated Frobenius powers on Y^{q^i} modulo the low degree HFE polynomial. Thus the complexity is something like $\mathcal{O}(nD)$.

A more serious issue is that the signing algorithm is not time constant. The authors acknowledge that the Cantor-Zassenhaus step, see [86], will only output an unique solution with probability approximately e^{-1}. Thus, for a full signature, there are roughly e^k calls to Cantor-Zassenhaus. Thus, there are likely some side-channel techniques to detect when solutions were not unique. The authors argue

Algorithm 2 GuiVerify

Input: $\mathcal{P}$, $\sigma = \mathbf{S}_k \| \mathbf{X}_k \| \cdots \| \mathbf{X}_1$ and $\mathbf{d}_i$ for $1 \leq i \leq k$.
Output: boolean b with value **TRUE** if signature accepted.

for all i **from** k **to** 1 **do**
  $\mathbf{w} \leftarrow \mathcal{P}(\mathbf{S}_i \| \mathbf{X}_i)$
  $\mathbf{S}_{i-1} \leftarrow \mathbf{w} \oplus \mathbf{d}_i$
end for
if $\mathbf{S}_0 = \mathbf{0}$ **then**
  return true
else
  return false
end if

that there are no known attacks for "big-field" schemes using the fact that an equation in the extension field does not have an unique solution.

Another issue with Gui is that the original submission did not adhere to the rules stated in the call for proposals. The submitters' implementation included the use of the PCLMULQDQ carry-less multiplication instruction set stating that most new Intel and AMD processors use this instruction set. There should be a significant performance difference when using this instruction set versus the platform independent version required by the NIST Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptograpy Standardization Process, see [87].

4.2 GeMSS

The GeMSS multivariate digital signature scheme submission can be found in [88]. GeMSS, as is Gui, is a "big-field" scheme in the HFEv$^-$ family. The GeMSS submission is very similar to that of Gui (and probably should have appeared before Gui in this manuscript, but the author began work on Gui first). The notation is slightly different, but the details of the implementation are very similar; consequently, we highlight the differences between the schemes.

The first difference in the schemes is the repetition factor in the Feistel-Patarin construction. While the original proposal for Gui suggests using $k = 2$ repetitions, GeMSS more conservatively suggests using $k = 4$ repetitions. This difference means that for similar parameters, GeMMS should be only 50% as fast (or perhaps more accurately, twice as slow) as Gui.

Another significant difference is that the inversion of the HFE polynomial in GeMSS uses the Berlekamp Algorithm, see [6], in place of Cantor-Zassenhaus, see [86]. Still, the expected complexity is about the same.

The final significant difference between GeMMS and Gui is also in relation to the Feistel-Patarin construction. The constrution requires the selection of a random "salt" of some specified length ℓ. In Gui, guidance was given in the selection of ℓ. Specifically, they suggest that a static value of $\ell = 128$ is used. GeMMS appears to provide no guidance in the selection of this value; thus, a

Parameter Set	q	n	r	v	a	k
GeMSS128	2	174	10	12	12	4
GeMSS192	2	265	10	20	22	4
GeMSS256	2	354	10	33	30	4

	Private Key Size	Public Key Size	Signature Size (w/ salt)
GeMSS128	13.9kB	407.6kB	384b
GeMSS192	38.5kB	1273.6kB	704b
GeMSS256	80.1kB	3519.3kB	832b

Fig. 6. The parameters of the GeMSS submissions and their private and public key sizes.

user is free to chose $\ell = 0$, which undermines the EUF-CMA proof. (In spite of this apparent oversight, the attention to detail of the GeMSS submission makes it seem more like the beginnings of a standards document than the Gui submission.)

The suggested parameters for standardization are provided in Figure 6.

4.3 HiMQ-3

HiMQ-3, standing for "High speed Multivariate Quadratic system with 3 layers," see [89], is a "small-field" scheme similar to Rainbow that uses specially structured layers to achieve a speed up over traditional Rainbow. The scheme first appeared at ASIACRYPT 2017 in [90]. The scheme utilizes a three layer Rainbow-like design with some components of the polynomial system of a special form that is efficiently solvable.

First, the submitters present the following lemma:

Lemma 1 *Let $\mathbb{F}$ be a finite field of characteristic 2 and let ℓ be an odd positive integer. Let $\mathcal{Q}_i(\mathbf{x}) = \alpha_i x_i x_{i+1} - \beta_i$ is $1 \leq i < \ell$ and let $\mathcal{Q}_\ell(\mathbf{x}) = \alpha_\ell x_\ell x_1 - \beta_\ell$, where $\alpha_i, \beta_i \in \mathbb{F}^*$. Let $A = \prod_{i=1}^{\ell} \alpha_i^{-1}\beta_i$, $B = \prod_{i=1}^{(\ell-1)/2} \alpha_i^{-1}\beta_i$. Then $\mathcal{Q}$ has an unique root $(x_1, \ldots, x_\ell)$ given by*

$$x_i = \begin{cases} \sqrt{A}/B, \text{ if } i = 1 \\ \alpha_{i-1}^{-1}\beta_{i-1}x_{i-1}^{-1}, \text{ if } 1 < i < \ell \\ \alpha_i^{-1}\beta_i x_1^{-1}, \text{ if } i = \ell. \end{cases}$$

Lemma 1 is used for efficient inversion of two layers while Gaussian elimination is used to solve the last layer. The structure of the central maps is best illustrated by the visualization in Figure 7

The design seems to be inspired by Rainbow with a more efficient inversion in some layers. The submitters utilize a trick similar to the ℓIC scheme, see [55], to streamline inversion while maintaining rank properties similar to an instance of Rainbow with the same parameters. To use this trick they require that the

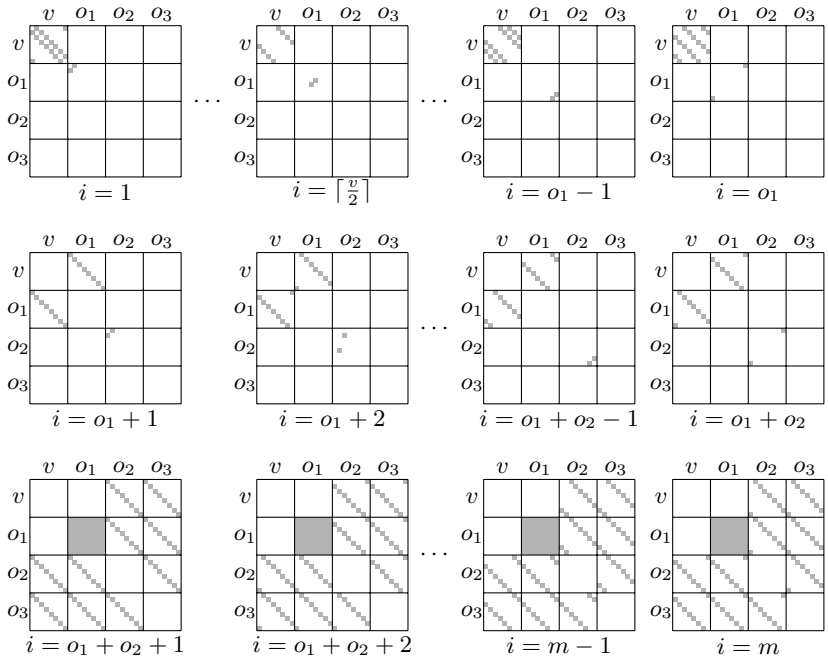


Fig. 7. The rough shape of the matrix representations of the discrete differential of each central map of HiMQ-3. The shaded regions represent possibly nonzero values while unshaded areas have coefficients of zero. (This diagram uses $v = o_1 = o_2 = o_3$ which is not a valid set of parameters for HiMQ-3, but makes the drawing simpler. For actual parameters, it is required that $v > o_1$.)

coefficients of the central map are non-zero as well as any valid input to the central map.

The submitters propose three versions of HiMQ-3 including a couple of variants of the idea. The first modification, called HiMQ-3F replaces the quadratic structure of the vinegar variables in the first layer with a random structure, and does the same to the monomials mixing the vinegar variables and first layer of oil variables in the second layer maps. The second modification is to generate the private key with a pseudo-random generator(PRG) to save storage space. The key specifications for each parameter set are listed in Figure 8.

4.4 LUOV

The LUOV submission, see [91], is a variant of the UOV scheme of [51]. The scheme incorporates several improvements on UOV, some of which have been around for years and some of which are new.

The first improvement LUOV offers over UOV is a key size reduction derived by requiring the coefficients of a map from a vector space $\mathbb{F}_q^n$ to $\mathbb{F}_q^m$ with a power

Parameter Set	q	v	o_1	o_2	o_3
HiMQ-3	256	31	15	15	14
HiMQ-3F	256	24	11	17	15
HiMQ-3P	256	31	15	15	14

	Private Key Size	Public Key Size	Signature Size
HiMQ-3	12074B	128744B	75B
HiMQ-3F	14878B	100878B	67B
HiMQ-3P	32B	128744B	75B

Fig. 8. The parameters of the HiMQ-3 submissions and their private and public key and signature sizes.

of 2 all lie in $\mathbb{F}_2 \subsetneq \mathbb{F}_q$, an idea first proposed in [92]. Defining UOV over $\mathbb{F}_2$ itself does not seem optimal because solving generic systems is slightly more efficient over $\mathbb{F}_2$ than over larger fields. This modification allows an adversary to consider the public key as a map from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$, though hash-and-sign seems to avoid these attacks in principal.

Another significant difference between LUOV and UOV is that *LUOV* uses the observation in [93] that many coefficients of the public key can be arbitrarily selected and a corresponding private key be derived to achieve a further key size reduction. While in [93], the authors suggest using a cyclic construction, LUOV simply publishes a seed for a PRG as part of the key.

Additional modifications include the introduction of a message recovery mode, an independent way of recovering some bytes of the message from the verification process, and the demonstration of a trade off between public key size and signature size. Thus, LUOV offers very malleable keys that can be tweaked for different performance and security properties.

The submitters provide six parameter sets, see Figure 9. Three of these proposals are for short signatures with larger keys and three are for longer signatures with shorter keys.

	Private Key Size	Public Key Size	Signature Size
LUOV-8-63-256	32B	15.5kB	319B
LUOV-8-90-351	32B	45.0kB	441B
LUOV-8-117-404	32B	98.6kB	521B
LUOV-48-49-242	32B	7.3kB	1.7kB
LUOV-64-68-330	32B	19.5kB	3.1kB
LUOV-80-86-399	32B	39.3kB	4.7kB

Fig. 9. The parameters of the LUOV submissions and their private and public key and signature sizes. The naming convention is LUOV-$log_2(q)$-o-v.

4.5 Rainbow

In [94], the Rainbow signature scheme is presented. Rainbow is one of the oldest and most studied "small-field" schemes, first appearing in [**?**]. In the lineage of UOV, Rainbow offers a potential peformance improvement due to smaller matrix inversion steps. Rainbow also offers a great deal of malleability with possible trade offs between key size and speed.

A detailed description of the Rainbow primitive is provided in Section 2.2. As described there, the scheme only achieves universal unforgeability. To achieve EUF-CMA security, the submitters recommend a transformation similar to the transformation in [85]. As in the case of Gui and GeMSS, this transform essentially replaces signing $\mathcal{H}(d)$ with signing $\mathcal{H}(\mathcal{H}(d)\|\mathbf{r})$ for a random "salt" $\mathbf{r}$.

Generically, given u layers with parameters $o_1, \ldots, o_u, v_1, \ldots, v_u$, m equations and n variables, the size of the private key is

$$m(m+1) + n(n+1) + \sum_{i=1}^{u} \left(\frac{v_i(v_i+1)}{2} + v_i o_i + v_i + o_i + 1 \right)$$

elements of $\mathbb{F}_q$. Meanwhile, the corresponding public key consists of

$$m\frac{(n+1)(n+1)}{2}$$

elements of $\mathbb{F}_q$.

The submission includes 9 parameter sets some of which are designed over odd characteristic fields as was promoted in [95] and some of which are over fields of characteristic 2 for efficiency. The Parameters are presented in Figure 10 while key sizes and signature sizes are offered in Figure 11.

Parameter Set	q	v_1	o_1	o_2	n	m
RainbowIa	16	32	32	32	96	64
RainbowIb	256	36	28	28	92	56
RainbowIc	256	40	24	24	88	48
RainbowIIIb	256	64	32	48	144	80
RainbowIIIc	256	68	36	36	140	72
RainbowIVa	256	56	48	48	152	96
RainbowVc	256	92	48	48	188	96
RainbowVIa	256	76	64	64	204	128
RainbowVIb	256	84	56	56	196	112

Fig. 10. The parameters of the Rainbow submissions.

The original Rainbow submission did not adhere to the rules stated in the call for proposals. The submitters' implementation included the use of the AVX2 instruction VPMADDUBSW which computes two 8-bit Single Instruction Multiple Data (SIMD) multiplications and a 16-bit SIMD addition in one operation.

There should be a significant performance difference when using this instruction set versus the platform independent version required by the NIST Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptograpy Standardization Process, see [87].

	Private Key Size	Public Key Size	Signature Size
RainbowIa	$97.9kB$	$148.5kB$	$512b$
RainbowIb	$103.7kB$	$148.3kB$	$624b$
RainbowIc	$140.0kB$	$187.7kB$	$832b$
RainbowIIIb	$371.4kB$	$512.1kB$	$896b$
RainbowIIIc	$525.2kB$	$703.9kB$	$1248b$
RainbowIVa	$367.3kB$	$552.2kB$	$736b$
RainbowVc	$1244.4kB$	$1683.3kB$	$1632b$
RainbowVIa	$781.2kB$	$1319.7kB$	$944b$
RainbowVIb	$922.4kB$	$1321kB$	$1176b$

Fig. 11. The private and public key sizes and signature sizes of the Rainbow submissions.

4.6 MQDSS

The MQDSS signature scheme, first published in [96], is based on the 5-pass multivariate identification scheme of [97] which is provably as hard as the $\mathcal{MQ}$ problem. The submission, see [98], uses a generalization of the Fiat-Shamir Tranform for 5-pass identification schemes on the identification scheme of [97].

The 5-pass identification scheme in [97] is based essentially on a zero-knowledge proof of knowledge of the space $R_F = \{(\mathbf{x}, \mathbf{y}) : \mathbf{y} = F(\mathbf{x})\}$, where F is a random multivariate quadratic system (in the context of MQDSS this polynomial has all constant coefficients zero). This proof exploits the fact that if $F(\mathbf{0}) = \mathbf{0}$, then $F(a + x) = DF(a + x) + F(a) + F(x)$ to split knowledge of the equation $\mathbf{y} = F(\mathbf{x})$ into two shares in the following way. First, notice that for any tuple $(r_0, r_1, t_0, t_1, e_0, e_1)$ the system

$$\begin{aligned} DF(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{e}_0 &= \mathbf{y} - F(\mathbf{r}_1) - DF(t_1, \mathbf{r}_1) - \mathbf{e}_1 \\ \mathbf{t}_0 &= \mathbf{r}_0 - \mathbf{t}_1 \\ \mathbf{e}_0 &= F(\mathbf{r}_0) - \mathbf{e}_1 \end{aligned}$$

is satisfied if and only if $\mathbf{y} = F(\mathbf{r}_0 + \mathbf{r}_1)$. So the prover chooses randomly $r_0, t_0 \in \mathbb{F}_q^n$ and $e_0 \in \mathbb{F}_q^m$ and sets $\mathbf{r}_1 = \mathbf{x} - \mathbf{r}_0$, $\mathbf{t}_1 = \mathbf{r}_0 - \mathbf{t}_0$ and $\mathbf{e}_1 = F(\mathbf{r}_0) - \mathbf{e}_0$, generating such a tuple from which the verifier can query a coordinate while receiving no knowledge of $\mathbf{x}$.

The scheme uses three pseudo-random generators (PRGs): PRG_{sk}, used to generate three pseudo-random seeds; PRG_s, used to generate a pseudo-random input for an $\mathcal{MQ}$ instance; and PRG_{rte}, a pseudo-random generator used during signing. The scheme also employs three cryptographic hash functions, $\mathcal{H}$,

H_1 and H_2, and two string commitment functions, Com_0 and Com_1. The secret key is simply a random seed for PRG_{sk}. The public key consists of the pseudo-random seed for an extendable output $\mathcal{MQ}$ encoder and the output of the resultant multivariate system on the pseudo-random input generated by PRG_s. Specifically, for security parameter k, we randomly choose $sk \in \{0,1\}^k$, parse $(S_F, S_s, S_{rte}) = PRG_{sk}(sk)$, generate the multivariate system $F = XOF_F(S_F)$ with differential $G(\mathbf{a}, \mathbf{x}) = DF(\mathbf{a}, \mathbf{x})$ (as specified, F has no constant coefficients, thus $F(0) = 0$), compute $s = PRG_s(S_s)$, evaluate $v = F(s)$ and form the public key $pk = (S_F, v)$. Thus, both the public key and private key are relatively small, see Figure 12.

Parameter Set	k	q	n	r	Private Key Size	Public Key Size	Signature Size
MQDSS-31-48	256	31	48	269	$62B$	$32B$	$32882B$
MQDSS-31-64	384	31	64	403	$88B$	$48B$	$67800B$

Fig. 12. Parameters and key sizes for MQDSS submissions.

To sign, one applies Algorithm 3 to obtain a signature which is essentially the concatenation of the hash of the concatenation of the secret key and message, a list of commitment shares, responses and a hash of the concatenation of all commitments. Verification is accomplished by shadowing the calculation, using the hash of the secret key and message to reconstruct the commitments. The details are found in Algorithm 4.

5 Security Analyses in Supporting Documentation

Of the submissions, two are "big-field" HFEv$^-$ schemes that are fairly similar, two are different approaches to the Rainbow construction, one is a new modifier for UOV, and one used a modified Fiat-Shamir transform on a provably hard identification scheme that embeds a transcript of a zero-knowledge proof in the signature. The security analyses of the HFEv$^-$-like schemes and those of the Rainbow-like schemes are similar, and the teams present the analysis in varying levels of detail. We describe the security results reported in the supporting documentation with the schemes grouped into their categories.

5.1 HFEv$^-$ Schemes

Both the Gui submission, see [82], and the GeMSS submission, see [88], acknowledge that there is no reduction theoretic proof of security available for HFEv$^-$ schemes. They both note, however, that HFEv$^-$ schemes are some of the oldest and best studied multivariate post-quantum cryptosystems, comparable in age to NTRU, see [99].

The teams discuss the resistance of HFEv$^-$ schemes to exhaustive search, the direct algebraic attack, to MinRank attacks, and the new distinguishing attack

Algorithm 3 MQDSSSign

Input: sk, M
Output: Signature σ

$S_F, S_s, S_{rte} \leftarrow PRG_{sk}(sk)$
$F \leftarrow XOF_F(S_F)$
$s \leftarrow PRG_s(S_s)$
$pk \leftarrow (S_F, F(s))$
$R \leftarrow \mathcal{H}(sk \| M)$
$D \leftarrow \mathcal{H}(pk \| R \| M)$
$r_0^{(1)}, \ldots, r_0^{(r)}, t_0^{(1)}, \ldots, t_0^{(r)}, e_0^{(1)}, \ldots, e_0^{(r)} \leftarrow PRG_{rte}(S_{rte}, D)$
for all j **from** 1 **to** r **do**
  $r_1^{(j)} \leftarrow s - r_0^{(j)}$
  $c_0^{(j)} \leftarrow Com_0(r_0^j, t_0^{(j)}, e_0^{(j)})$
  $c_1^{(j)} \leftarrow Com_1(r_1^{(j)}, G(t_0^{(j)}, r_1^{(j)}) + e_0^{(j)}$
  $com^{(j)} \leftarrow (c_0^{(j)}, c_1^{(j)})$
end for
$\sigma_0 \leftarrow \mathcal{H}(com^{(1)} \| com^{(2)} \| \ldots \| com^{(r)})$
$ch_1 \leftarrow H_1(D, \sigma_0)$
Parse ch_1 as $ch_1 = (\alpha^{(1)}, \ldots, \alpha^{(r)} \in \mathbb{F}_q^r$
for all j **from** 1 **to** r **do**
  $t_q^{(j)} \leftarrow \alpha^{(j)} r_0^{(j)} - t_0^{(j)}$
  $e_1^{(j)} \leftarrow \alpha^{(j)} F(r_0^{(j)}) - e_0^{(j)}$
  $resp_q^{(j)} \leftarrow (t_1^{(j)}, e_1^{(j)})$
end for
$\sigma_1 \leftarrow (resp_1^{(1)} \| resp_1^{(2)} \| \ldots \| resp_1^{(r)})$
$ch_2 \leftarrow H_2(D, \sigma_0, ch_1, \sigma_1)$
Parse ch_2 as $ch_2 = (b^{(1)}, \ldots, b^{(r)}) \in \mathbb{F}_q^r$
for all j **from** 1 **to** r **do**
  $resp_2^{(j)} \leftarrow r_{b^{(j)}}^{(j)}$
end for
$\sigma_2 \leftarrow (resp_2^{(1)} \| \ldots \| resp_2^{(r)} \| c_{1-b^{(1)}}^{(1)} \| \ldots \| c_{1-b^{(r)}}^{(r)})$
return $\sigma = (R, \sigma_0, \sigma_1, \sigma_2)$

Algorithm 4 MQDSSVerify

Input: pk, σ, M
Output: boolean b with value **TRUE** if signature accepted.

$F \leftarrow XOF_F(S_F)$
$D \leftarrow \mathcal{H}(pk \| R \| M)$ assuming σ parsed as $(R, \sigma_0, \sigma_1, \sigma_2)$
$ch_1 \leftarrow H_1(D, \sigma_0)$
Parse ch_1 as $ch_1 = (\alpha^{(1)}, \ldots, \alpha^{(r)} \in \mathbb{F}_q^r$
$ch_2 \leftarrow H_2(D, \sigma_0, ch_1, \sigma_1)$
Parse ch_2 as $ch_2 = (b^{(1)}, \ldots, b^{(r)}) \in \mathbb{F}_q^r$
Parse σ_0 as $(resp_1^{(1)} \| resp_1^{(2)} \| \ldots \| resp_1^{(r)})$
Parse σ_2 as $(resp_2^{(1)} \| \ldots \| resp_2^{(r)} \| c_{1-b^{(1)}}^{(1)} \| \ldots \| c_{1-b^{(r)}}^{(r)})$
for all j **from** 1 **to** r **do**
  Parse $resp_q^{(j)}$ as $(t_1^{(j)}, e_1^{(j)})$
  if $b^{(j)} = 0$ **then**
    $r_0^{(j)} \leftarrow resp_2^{(j)}$
    $c_0^{(j)} \leftarrow Com_0(r_0^j, \alpha^{(j)} r_0^{(j)} - t_1^{(j)}, \alpha^{(j)} F(r_0^{(j)} - e_1^{(j)})$
  else
    $r_1^{(j)} \leftarrow resp_2^{(j)}$
    $c_1^{(j)} \leftarrow Com_1(r_1^{(j)}, \alpha^{(j)}(v - F(r_1^{(j)})) - G(t_1^{(j)}, r_1^{(j)}) - e_1^{(j)}$
  end if
  $com^{(j)} \leftarrow (c_0^{(j)}, c_1^{(j)})$
end for
$\sigma_0' \leftarrow \mathcal{H}(com^{(1)} \| \ldots \| com^{(r)})$
return $(\sigma_0' = \sigma_0)$

in [75]. In addition, both teams note the known quantum speedups for these algorithms. In particular, it is noted that there is no known quantum speedup for the rank attacks and appears to only be potential for a quantum speedup of direct attacks with the hybrid methodology.

First, we consider an exhaustive search message recovery attack. The fastest known method for a quantum adversary for inverting a quadratic system over $\mathbb{F}_2$ is Quantum FXL, see [77]. The complexity for HFEv$^-$ parameters is

$$\mathcal{O}\left(2^{0.45(n-a)}\right).$$

If we cautiously assume we can apply the same technique in general, we can replace the 2 in the above formula with q. (In particular, we can make this exchange anyway in characteristic 2.)

We next consider the direct algebraic attack, a message recovery attack based on algebraic techniques for solving a system of nonlinear equations typically by way of a Gröbner basis calculation. There has been a great deal of work in the literature on deriving the degree of regularity of schemes related to HFE. A progression of results deriving upper bounds on the degree of regularity, first by assuming the system to be semi-regular and subsequently by using the special structure of HFE, can be found in [100, 79, 101, 80, 81]. Empirical evidence of the lower bound trendline for the degree of regularity of HFEv$^-$ schemes can be found in [102]. Both teams assert that the culmination of these works seems to provide the generic complexity of the algebraic approach. The empirical lower bound provides a complexity against direct attacks of

$$\mathcal{O}\left(\binom{n-a}{\lceil\frac{r+a+v+7}{3}\rceil}^{\omega}\right),$$

where $2 \leq \omega \leq 3$ is the linear algebra constant. Considering that both schemes use a Feistel-Patarin construction with a repetition factor, it is not clear how to implement a hybrid attack with a guessing component to exploit a quantum speedup from Grover's algorithm.

We also consider the MinRank attack, that breaks HFE, see [70], and weak parameters of HFE$^-$, see [15]. This attack exploits the property imposed by the degree bound of HFE maps that they have a low Q-rank, that is, a low rank as a quadratic form on a $\mathbb{K}$-algebra $\mathbb{A}$. This attack, first proposed in [39], was updated in [70] to a var more efficient attack. The low rank property can be encoded in a system of equations over $\mathbb{F}$, a Gröbner basis computed over $\mathbb{F}$, and the variety computed over $\mathbb{K}$ to recover a low rank representation of the central map as a quadratic form on $\mathbb{A}$. While minors modeling currently has the tightest complexity analysis, Kipnis Shamir modeling seems experimentally to be slightly better for these schemes. Both teams report the complexity to be

$$\mathcal{O}\left(n^{\omega(r+v+a+1)}\right).$$

While the Gui team use a value of $\omega = 2.3$ for setting parameters, the GeMSS team choose $\omega = 2$. For the size of schemes presented, it seems that both choices of ω are conservative, especially the GeMSS selection.

Differential attacks have also been a concern for schemes in the lineage of C^*. The teams both cite the result in [14] proving HFEv$^-$ schemes immune to differential cryptanalysis.

Finally, both teams acknowledge the recent statistical approach of [75] combining random projections with MinRank and hybrid algebraic attacks on the primitive. For all of the parameter sets, the strongest of these statistical techniques is the hybrid algebraic distinguishing attack. The complexity is given by

$$\mathcal{O}\left(2^{(n-k)/2}\binom{n+v-k}{d_{reg}}^{\omega}\right),$$

for the quantum adversary.

Since the two schemes are very similar with a similar methodology in the selection of parameters, they achieve similar security properties with respect to known attack methods. A summary of these results is in Figure 13.

Parameter Set	Exhaustive Search	Direct Algebraic	MinRank	Distinguishing
Gui-184	**108**	157	324	**191**
Gui-312	**171**	222	481	**280**
Gui-448	**236**	293	665	**382**
GeMSS128	**105**	131	513*	⋆
GeMSS192	**147**	193	824*	⋆
GeMSS256	**189**	261	1218*	⋆

Fig. 13. The log_2 of the Complexity of known attacks **quantum** or classical on HFEv$^-$ submissions.
* - derived from a weak asymptotic approximation.
⋆ - not available in submission.

5.2 Rainbow-like Schemes

The submissions HiMQ-3, LUOV, and Rainbow can all be classified as Rainbow schemes with different parameters and possibly extra structure. They can all be attacked by rank methods, and algebraic attacks. HiMQ-3 and LUOV have additional structure that given them extra properties.

In applying the direct algebraic attack, the adversary is free to arbitrarily assign values to enough variables to achieve a fully determined system reducing the number of variables to equal the number of equations. In addition to this, the adversary can employ the hybrid approach in which a few more values of variables are assigned at random. If the guess is correct, then the system is slightly overdetermined and may result in the degree of regularity lowering, so that the resolution is faster. A quantum implementation of this hybrid approach allows the search to be sped up by Grover's algorithm, resulting in further improvement.

While Rainbow has been studied for years in the context of algebraic attacks, see [103], for example, both HiMQ-3 and LUOV are relatively new and this area is

less explored. The HiMQ-3 addresses this issue by running experiments showing that the systems appear to be semi-regular as is optimal. The LUOV team notes that only a portion of the publically computable coefficients of the public map lie in $\mathbb{F}_2$ while the rest lie in $\mathbb{F}_8$. Thus, direct attacks must solve over the larger field. They further state that the work in [104] suggests that the systems arising from UOV by assigning values to v variables (as is performed in the fastest attacks) appear semi-regular. The team concludes that the direct algebraic attack is of the same complexity as an instance of UOV with similar parameters.

The low rank attack has a complexity determined by the ratio of equations to variables, and since these schemes are underdefined, the complexity of the linear algebra search version of this attack is $q^r m^\omega$, where r is equal to the number of oil variables in the first layer and m is equal to the total number of oil variables. In the case of LUOV, these two quantities are equal and smaller than v, so we do not expect to find low rank maps in the span of the public quadratic forms. In the case of HiMQ-3, the central maps have very few monomials, but are required to have full rank. For both HiMQ-3 and Rainbow the complexity of linear algebra search is $\mathcal{O}\left((v+o_1+o_2)^\omega q^{(v+1)/2}\right)$ for a quantum adversary due to the interlinked kernels, see [105].

The high rank attack and Rainbow Band Separation(RBS) attack are also applicable to both Rainbow and HiMQ-3. The attack find the span of the last oil layer which consists of the variables occuring the least times in quadratic terms in the central map. LUOV is immune to the high rank attack; the RBS attack can in principle be used to attack LUOV. The complexity for a quantum adversary for this attack is $\mathcal{O}\left((v+o_1+o_2)^\omega q^{o_2/2}\right)$.

A summary of the security analysis for these scheme is presented in Figure 14. We focus on some of the parameters instead of every iteration for convenience.

Parameter Set	Exhaustive Search	Direct Algebraic	MinRank	HighRank	RBS
HiMQ-3	**154**	129	131	135	$\star$
RainbowIa	**115**	**147**	**95**	**86**	145
LUOV-8-63-256	226	161	UOV$\rightarrow$	**161**	192

Fig. 14. The log_2 of the Complexity of known attacks **quantum** or classical on Rainbow-like submissions.
$\star$ - not available in submission.

5.3 MQDSS

MQDSS differs from the other submissions in that it comes with a provable reduction to $\mathcal{MQ}$. There are some issues to consider still.

First, the parameters selected do not satisfy the requirements of the security proof. In particular, in [98, Section 11.1], the submitters state that, "our security reduction in the ROM is very loose, we found it impractical to use concrete expressions from the reduction in our choice of parameters. Instead, the

parameters are based on the best known attacks agains the $\mathcal{MQ}$ problem and against SHAKE256." Furthermore, there may be problems with the extension of the result in the QROM model. The submitters themselves note this in Section 14 of the submission.

The parameters of MQDSS-31-48 provide 99 bits of security against quantum exhaustive search and 159 bits of security against classical Gröbner basis algorithms.

6 Performance Comparison

We summarize the performance of the smallest version of each scheme in terms of key size in Figure 16 and in terms of speed in Figure **??**.

Parameter Set	Private Key Size	Public Key Size	Signature Size
Gui-184	19.1kB	416.3kB	360b
GeMSS128	13.9kB	407.6kB	384b
HiMQ-3	11.8kB	125.7kB	75B
RainbowIa	97.9kB	148.5kB	64B
LUOV-8-63-256	32B	15.5kB	319B
MQDSS-31-48	62B	32B	32.1kB

Fig. 15. A comparison of key and signature sizes among the multivariate candidates.

Parameter Set	KeyGen	Sign	Verify
Gui-184	2408M	1910M	152k
GeMSS128	44ms	323ms	41μs
HiMQ-3	50.6M	21.6k	18.0k
RainbowIa	1302M	601k	350k
LUOV-8-63-256	21M	5.9M	4.9M
MQDSS-31-48	1.2M	52.5M	38.7M

Fig. 16. A comparison of cycles among the multivariate candidates. Not available for GeMSS

7 Recent Work and Comments

In this section we present the current official comments on the multivariate submissions.

From: Ward Beullens <ward.beullens@student.kuleuven.be>
Sent: Monday, April 30, 2018 6:41 AM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: Gui

Dear all,

In my previous email I forgot to include the references, here they are:

[1] Nicolas Courtois. Generic attacks and the security of quartz. In Public Key Cryptography, volume 2567 of Lecture Notes in Computer Science, pages 351–364. Springer, 2003.
[2] Van Oorschot, Paul C., and Michael J. Wiener. "Parallel collision search with cryptanalytic applications." *Journal of cryptology* 12.1 (1999): 1-28.

My apologies,
Ward

On 04/27/2018 04:11 PM, Ward Beullens wrote:

> Dear all,
>
> I believe there is a problem with the parameters of the Gui signature scheme for security level 1, and that a parameter change is needed.
>
> The scheme uses a HFEv- trapdoor function which, with the proposed parameters for security level 1, outputs 168 bits. Given the limited number of output bits, this trapdoor cannot be straightforwardly used in a hash-and-sign scheme, because a collision attack would be able to forge signatures with roughly 2^{168/2} = 2^84 evaluations of the trapdoor function. Instead, Gui uses the Feistel-Patarin construction [1], which requires k inversions of the trapdoor function to sign a message and k evaluations of the trapdoor function to verify a signature.
>
> The paper [1] describes a generic attack on the Feistel-Patarin construction with requires roughly 2^{m*k/k+1} evaluations of the trapdoor function (where m is the number of bits outputted by the trapdoor function), and requires roughly m*2^{m*k/k+1} bits of memory. For Gui this means 2^112 evaluations of the public map, and 112*2^112 bits of memory.
>
> However, the distinguished point method of [2] can be used to have essentially the time complexity with roughly 3*112*2^56 bits of memory (that is less than the amount of data that Google stores). I estimate that this attack requires 2^135 (classical) gates, which is significantly less than the estimate of 2^143 gates for a key-search on AES in the NIST call for proposals.
>
> I think the best way to fix the problem is to increase the parameter k from 2 to 3 (the GeMSS submission has similar parameters and uses k=4). This would lead to a very modest increase of 32 bits in signature size, and a slowdown of the signing and verification algorithm of 50%.
>
> I want to stress that this is a purely generic attack which only affects the security level 1 parameters, this does not indicate a weakness in the HFEv- construction.
>
> Kind regards,
> Ward

From: Bo-Yin Yang <moscito@gmail.com>
Sent: Thursday, June 14, 2018 9:33 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: Gui

Dear Ward and everyone on this list,

We agree that we made a small mistake in our parameters and will change from k=2 to k=3 in Gui-184 in the future. This will not affect keysizes but will increase the signature by 32 bits as well as the runtime by 1.5x.

Best wishes
The Gui designers

From: D. J. Bernstein <djb@cr.yp.to>
Sent: Monday, April 30, 2018 3:45 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: GeMSS
Attachments: signature.asc

I'm writing to correct some misimpressions regarding asymptotic MQ security that are created by the GeMSS/DualModeMS presentation:

https://csrc.nist.gov/CSRC/media/Presentations/DualModeMS-GeMMS/images-media/DualMode-and-GeMMs-April2018.pdf

Specifically, page 15 of the PDF (slide "10/19") has the following summary of the costs of quantum versions of FXL stated in two independent papers for solving m equations in m variables over F_q:

* "O(2^0.462m)" from a "2017" paper (posted 2017.12.19).
* "When q=2, O(2^0.472m)" from a "PQC 2018" paper (posted 2017.12.15).

There are four specific issues here.

Issue #1: Because the "When q=2" restriction is stated only for the second number, readers will assume that the first number applies to larger fields---for example, that switching from F_2 to F_3 for the same m doesn't provide larger security at this level of detail.

That's wrong. The first number 0.46240... is also limited to F_2. For
F_3 the best exponent known is 0.70425..., below 0.5 lg 3 = 0.79248...
but above 0.46240..., analogously to the pre-quantum situation of F_3 having larger exponent than F_2. See the 2017.12.15 paper for details.
(The 2017.12.19 paper doesn't consider cases beyond F_2.)

Issue #2: Readers will assume that the 0.462 and 0.472 are the best exponents obtained in these two papers, and are likely to think that this discrepancy shows some instability in the understanding of this class of algorithms.

But that's also wrong. The two papers obtain the same exponent here. See Table 4.10 in the 2017.12.15 paper, "GroverXL operation-count exponent ... rounded down to multiple of 0.00001", top-left corner (top being F_2; left being the same number of equations as variables), "0.46240".
This is the same as the "0.462" exponent from the 2017.12.19 paper.

There's also a "0.47210" in the 2017.12.15 paper, but that's in a different metric, so it's wrong to juxtapose the numbers without mentioning that the metrics are different. Specifically:

* Exponent 0.46240...+o(1) is in a simplified operation-count metric. This metric is considered in both papers.

* Exponent 0.47210...+o(1) is in a realistic area-time metric. This metric is analyzed only in the 2017.12.15 paper, and this exponent isn't achieved by the algorithm outlined in the 2017.12.19 paper.

The gap here occurs for the same reasons as the long-established gap between analogous metrics for pre-quantum integer factorization: linear algebra uses a lot of communication.

Issue #3: The "0.462" and the "0.472" are the results of rounding the actual exponents down. This needs to be stated explicitly, for example with dots. The issue here isn't that this is a big quantitative gap; the issue is that careful readers comparing, e.g., "0.462" to "0.46240..."
are again being told that there's a discrepancy, which isn't true.

Issue #4: A o(1) in the exponent has disappeared in favor of an O() outside the formula. This isn't justified by either paper. This _could_ be a big quantitative gap compared to any reasonable O constant---one would have to do a more detailed analysis to tell.

Of course rounding _up_ can avoid the overt error: if the time is at most 2^((0.462...+o(1))m) then it's true that the time is at most, say, O(2^(0.463m)). However, careful readers comparing two of these slight asymptotic overestimates are again led to believe that there's a discrepancy when there actually isn't (unless the slight overestimates happen to coincide). Furthermore, the o(1) is useful as an alert regarding suppressed subexponential factors.

---Dan

P.S. I'm a coauthor of the 2017.12.15 paper and gave a talk on the paper at PQCrypto 2018. I was careful in the talk to point out the subset that was done independently (obtaining the same 0.46240... exponent) in the 2017.12.19 paper. I'm puzzled that the authors of the 2017.12.19 paper, overlapping the authors of these slides, have chosen to juxtapose 0.472 from "2018" with 0.462 from "2017" without mentioning that the metrics are different, without mentioning that the "2018" paper also obtained the 0.462 result for the smaller metric, and without mentioning that the "2018" paper was posted before the "2017" paper was.

From: perret <ludovic.perret@lip6.fr>
Sent: Tuesday, May 08, 2018 6:57 AM
To: pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: GeMSS

Dear Mailing list,

We have 20 minutes to present GeMSS and DualModeMS. Contrarily to the PQC'18 presentation on:
Daniel J. Bernstein and Bo-Yin Yang: ``Asymptotically faster quantum algorithms to solve multivariate quadratic equations'', the goal of the GeMSS/DualModeMS talk was not to have an in-depth discussion about recent results on the asymptotic hardness of MQ in the quantum setting. We tried to give a global idea on the features of GeMSS & DualModeMS and explain our strategy to derive the parameters.

The purpose of the slide that is pointed by D. Bernstein was too explain the strategy for evaluating the security of GeMSS/DualModeMS in the quantum setting. In particular, we wanted to emphasize that we already used O(2^0.462m) in the reference documentation of GeMSS/DualModeMS presentation. For the record, we copy-paste below the related paragraph in the GeMMS documentation submitted to NIST (deadline was end of November 2017).
``
QuantumBooleanSolve. In a recent paper [35], the authors present a quantum version of BooleanSolve that takes advantages of Grover's quantum algorithm [44]. QuantumBooleanSolve is a Las-Vegas quantum algorithm allowing to solve a system of m boolean equations in m variables. It uses O(n) qbits, requires the evaluation of, on average, O(2^0.462m) quantum gates. This complexity is obtained under certain algebraic assumptions.
''
where [35] is:
Jean-Charles Faugère, Kelsey Horan, Delaram Kahrobaei, Marc Kaplan, Elham Kashefi, and Ludovic Perret. Fast quantum algorithm for solving multivariate quadratic equations. To appear.

We have no doubt that the PQC'18 paper was done independently. However, this paper was only available 2017.12.15; so after NIST's submission deadline.

In fact, [35] was submitted to PKC'18 and rejected with quite unfair reasons to our point of view. Anyway, the situation is as it is. [35] was only made publicly available after the PQC'18 paper.

The current version of [35] is available here: https://eprint.iacr.org/2017/1236.pdf
It is currently under revision. We will inform the list as soon as the paper is available. This will be a better basis for comments; rather than (over)interpreting the slides on GeMSS/DualModeMS; that are somewhat unrelated to the issue.

Best Regards,

Ludovic Perret, on the behalf of the authors of [35] (Jean-Charles Faugère, Kelsey Horan, Delaram Kahrobaei, Marc Kaplan, Elham Kashefi)

> Le 30 avr. 2018 à 09:44, D. J. Bernstein <djb@cr.yp.to> a écrit :
>
> I'm writing to correct some misimpressions regarding asymptotic MQ
> security that are created by the GeMSS/DualModeMS presentation:

From: Ward Beullens <ward@beullens.com>
Sent: Wednesday, May 02, 2018 4:42 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: HiMQ-3

Dear all,

TL;DR: The security proof of HiMQ-3 (Theorem 4) is flawed.

The HiMQ-3 submission document claims that the HiMQ-3 signature scheme is EUF-CMA secure provided that it is hard to find a solution for a system of quadratic equations in the HiMQ-3 family. In other words, the claim is that if the scheme is UF-KOA secure (univeral forgery under key-only attack), then the scheme is also EUF-CMA secure.

The proof of this claim is to be found in [1] (Theorem 4.1), where the same claim is made for the ELSA signature scheme. The proof is very similar to the classic proof of [2] for the security of a hash-and-sign signature scheme based on a trapdoor permutation. However, the trapdoor function used by the HiMQ-3 scheme is not a permutation, and this causes the proof to fail.

The proof programs a random oracle by sampling random x, and returning P(x), where P is the public key. In the trapdoor permutation setting this is a valid approach, because there is no way to distinguish (x,P(x)) from (P^{-1}(y),y), for x and y uniformly distributed variables on the domain and codomain of P respectively. When P is no longer a permutation (as is the case for HiMQ-3 and ELSA) this might no longer be the case. (In fact, P^{-1}(y) is not even uniquely defined) This means that the adversary is no longer guaranteed to function correctly in the simulated environment and that the proof fails.

Kind regards,
Ward

[1] Shim, Kyung-Ah, Cheol-Min Park, and Namhun Koo. "An Existential Unforgeable Signature Scheme Based on Multivariate Quadratic Equations." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2017.

[2] Bellare, Mihir, and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols." Proceedings of the 1st ACM conference on Computer and communications security. ACM, 1993.

From: Ryo Fujita <rfujita140411@gmail.com>
Sent: Wednesday, July 18, 2018 1:58 AM
To: pqc-forum
Cc: pqc-comments
Subject: Re: OFFICIAL COMMENT: HiMQ-3

EUF-CMA security on multivariate signature schemes was discussed in [3]. There, it is described how to modify the signature scheme to achieve EUF-CMA in the random oracle model. Likewise, it seems that HiMQ-3 may also achieve EUF-CMA.

Kind regards,
Ryo

[3] Sakumoto K., Shirai T., Hiwatari H. (2011) On Provable Security of UOV and HFE Signature Schemes against Chosen-Message
Attack. In: Yang BY. (eds) Post-Quantum Cryptography. PQCrypto 2011. Lecture Notes in Computer Science, vol 7071. Springer, Berlin, Heidelberg

2018年5月3日木曜日 5時42分28秒 UTC+9 Ward Beullens:

> Dear all,
>
> TL;DR: The security proof of HiMQ-3 (Theorem 4) is flawed.
>
> The HiMQ-3 submission document claims that the HiMQ-3 signature scheme is EUF-CMA secure provided that it is hard to find a solution for a system of quadratic equations in the HiMQ-3 family. In other words, the claim is that if the scheme is UF-KOA secure (univeral forgery under key-only attack), then the scheme is also EUF-CMA secure.
>
> The proof of this claim is to be found in [1] (Theorem 4.1), where the same claim is made for the ELSA signature scheme. The proof is very similar to the classic proof of [2] for the security of a hash-and-sign signature scheme based on a trapdoor permutation. However, the trapdoor function used by the HiMQ-3 scheme is not a permutation, and this causes the proof to fail.
>
> The proof programs a random oracle by sampling random x, and returning P(x), where P is the public key. In the trapdoor permutation setting this is a valid approach, because there is no way to distinguish (x,P(x)) from (P^{-1}(y),y), for x and y uniformly distributed variables on the domain and codomain of P respectively. When P is no longer a permutation (as is the case for HiMQ-3 and ELSA) this might no longer be the case. (In fact, P^{-1}(y) is not even uniquely defined) This means that the adversary is no longer guaranteed to function correctly in the simulated environment and that the proof fails.
>
> Kind regards,
> Ward
>
> [1] Shim, Kyung-Ah, Cheol-Min Park, and Namhun Koo. "An Existential Unforgeable Signature Scheme Based on Multivariate Quadratic Equations." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2017.
>
> [2] Bellare, Mihir, and Phillip Rogaway. "Random oracles are practical: A paradigm for designing efficient protocols." Proceedings of the 1st ACM conference on Computer and communications security. ACM, 1993.

From: LOUISY Anne-Elise <anne-elise.louisy@thalesgroup.com>
Sent: Tuesday, August 07, 2018 10:38 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: HiMQ-3

Dear HiMQ-3 team,

There seem to be contradictions between the description of the third layer of the central map and the matrices presented in the analysis of known attacks (figure 2 of the supporting documentation).

In the description, it is written that the polynomials of the third layer of the central map are of the form:

f(x)= sum_i,j beta_i,j x_i x_j + theta(x) + theta'(x) + epsilon x_(o1+o2+k)

where the i,j in the sum are between v +1 and v1.

For the definition of theta and theta', it is written that the coefficients are such that symmetric matrix associated to the quadratic part of f has full rank, which implies that the quadratic part of f involves all n variables.

However, not all variables appear in f. For the k-th polynomial of the third layer, x_(v+1),…,x_(v1) appear in the sum, x_(v1+1),..,x_v2 appear in theta (assuming the modulo is o2 and that 1 is added to the result) and x_(v2+1),…, x_n appear in theta' (assuming again that 1 is added to the subscript). All the other variables, save for x_k that appears in theta and theta', are not in f.

Moreover, with the definition of the third layer given in the description of the central map, we get matrices with non-zero coefficients only in the square corresponding to the sum and on line k and column k resulting from theta and theta' (x_k appear in the products x_k x_i for several different i between v1+1 and n).

(the theoretical secret key size provided also suggests that they are more coefficients that the one given in the description),

Sincerely,

A-E. Louisy,

Student in cryptography at Versailles University

From: 심경아 <shimkah221@gmail.com>
Sent: Thursday, September 13, 2018 5:13 AM
To: pqc-comments
Subject: OFFICIAL COMMENT: HiMQ-3

Dear A-E. Louisy,

Thank you for your comments.

There is a typo. The current formulas

\Theta_i(x)=\sum_{j=1}^{v_1} \gamma_{i,j}x_i x_{v_1+(i+j-1) (mod o_3),

\Theta_i'(x)=\sum_{j=1}^{v_2} \gamma_{i,j}'x_i x_{v_2+(i+j-1) (mod o_3)

should be changed to

\Theta_i(x)=\sum_{j=1}^{v_1} \gamma_{i,j}x_j x_{v_1+(i+j-1) (mod o_2),

\Theta_i'(x)=\sum_{j=1}^{v_2} \gamma_{i,j}'x_j x_{v_2+(i+j-1) (mod o_3),

Note that $1 \le A (mod B) \le B$ for an integer A and a positive integer B,

in our definition.

Kind regards

Kyung-Ah Shim

****

Answer to Our Security Proof.

Due to the use of the multivariate quadratic map requiring additional random Vinegar variables, our trapdoor function is not permutation and the signature distribution is not uniformly distributed as presented in [1]. The authors [1] make the distribution of signatures uniform by using a random salt to the message being hashed and re-choosing a random salt instead of Vinegar variables.

We can use the same way to prove unforgeability of our scheme. For it, we need to propose a modified version: the modified signing algorithm is the same as the original one except that

-choose a random r \in {0, 1}^R, compute H(m, r)=h.

-If one of the linear systems has no signature then choose another random r’ and try again.

-Then the signature is (\tau, r).

In Verify algorithm, to verify a signature (\tau, r) on a message m, check whether the equation \cal P(\tau)=H(m, r) holds or not.

In the security proof, the H-query should be changed as:

For H-queries, the tuples in H-list are of the form (m_i, c_i, \tau_i, r_i, P(\tau_i)). When {\cal A} queries H at m_i \in \{0, 1\}^*,

i) If the query already appears on H-list in a tuple (m_i, c_i, \tau_i, r_i, P(\tau_i)) then {\cal} B returns H(m_i, r_i)=P(\tau_i).

ii) Otherwise, {\cal B} picks a random coin c_i \in {0,1} with Pr[c_i=0]=\frac{1}{q_S+1}.

-If c_i=1 then {\cal B} chooses a random \tau_i \in F_q^n and r_i \in {0, 1}^R, adds a tuple (m_i, c_i, \tau_i, r_i, P(\tau_i)) to H-list and returns H(m_i, r_i)=P(\tau_i).

- If c_i=0 then {\cal B} adds (m_i, c_i, r^*, *, \eta) to H-list from the instance and returns H(m_i, r^*)=\eta, where \eta is the given MQ-instance.

For Sign Queries. When {\cal A} makes a Sign-query on m_i, \cal B finds the corresponding tuple (m_i, c_i, \tau_i, r_i, P(\tau_i)) from H-list.

-If c_i=1 then \cal B responds with (\tau_i, r_i).

-If c_i=0 then \cal B reports failure and terminates.

Then the distribution of the outputs H(m_i, r_i) of our random oracle is identical to the distribution of \cal P(\tau), \tau \in_R F_q^n, since \tau is uniformly distributed over F_q^n and it is a valid signature satisfying \cal P(\tau)=H(m, r).

The rest of the proof is the same as that in [2].

[1] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari, On Provable Security of UOV and HFE Signature Schemes against Chosen-Message Attack, PQCrypto 2011, LNCS 7071, pp. 68–82, 2011.

[2] Kyung-Ah Shim, Cheol-Min Park, Namhun Koo: An Existential Unforgeable Signature Scheme Based on Multivariate Quadratic Equations. ASIACRYPT (1) 2017: pp. 37-64, 2017.

From: simona s <simona.samardziska@gmail.com>
Sent: Monday, September 03, 2018 8:11 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: MQDSS

Dear all,

Recently, after a related inquiry by Eliane Koussa and Ludovic Perret, we noticed that we have made a mistake in the choice of parameters in the NIST submission of MQDSS. In particular, the number of rounds in the submission is twice bigger than it is actually needed for the respective security level. This means that the number of rounds can be halved without affecting the security of the scheme, while substantially improving its performance:
the signing and verification time will be halved, and (even more importantly,) the signature size will be halved.

We therefore announce a new Version 1.1 of MQDSS, in which this mistake has been corrected.
The specification and implementation of MQDSS Version 1.1. are available through our (brand new) web site http://mqdss.org
(Direct link to specification: http://mqdss.org/files/MQDSS_Ver1point1.pdf
and to reference implementation https://github.com/joostrijneveld/MQDSS/tree/NIST)

The new parameters of MQDSS give the following performance results:

	Public key (bytes)	Secret key (bytes)	Signature (KiB)
MQDSS-31-48	46	16	16.15
MQDSS-31-64	64	24	33.23

Reference implementation:

	keygen (cycles)	signing (cycles)	verification (cycles)
MQDSS-31-48	1302K	26500K	19674K
MQDSS-31-64	2769K	84615K	63210K

Implementation using AVX2 instructions:

	keygen (cycles)	signing (cycles)	verification (cycles)
MQDSS-31-48	1078K	3683K	2504K
MQDSS-31-64	2495K	8709K	6183K

We respectively hope that NIST will take into account the new parameters of MQDSS for the first round of the PQC standardization process, especially since they only improve the performance of the scheme (the security remains the same).
We also understand that NIST has the right to evaluate the candidates based solely on the initial submission.

Sincerely,
The MQDSS team

8 Summary and Selection Criteria Discussion

The performance in terms of speed of multiple multivariate submissions suffer because of the requirement that the submission be in ANSI C with no assembly or platform dependent vector instructions. It is certainly natural to have these instruction sets for someone working in this area. Something that might be interesting to consider is how appropriate the restriction may seem for a fair comparison depending on the target application. We might have different opinions if we are selecting schemes to secure the internet versus an application in a sensor network.

Speed seems to be an issue with a few of these schemes. Key size is possibly even a greater burden for a couple of these schemes. On the other hand, there are some novelties in the submissions affecting key size.

HiMQ-3 requires the coefficients to be nonzero and if a variable takes the value zero then the efficient inversion collapses. Also, it is not clear from the analysis of equivalent keys whether the submitters considered the preservation of this property under the key transforms.

LUOV has a very interesting pair of modifications to reduce key size at the cost of efficiency. The property that some of the public key can be selected randomly has been known for almost a decade, but the field lifting idea is just a year old.

Three of the submissions are simply parameter selections for two cryptosystems that are twenty years old and extremely well studied. We can have a great deal of confidence in the schemes and the parameters, though far more agressive for the HFEv$^-$ schemes, seem to be both competitive with other parametrizations in the literature as well as solid.

The one submission in this group with a proof of security, MQDSS, selects parameters invalidating the proof. In fact, the scheme might have had comparable performance to its peers if the submitters had adhered to the parameter restrictions the proof required.

References

1. Huang, Y., Liu, F., Yang, B.: Public-key cryptography from new multivariate quadratic assumptions. In Fischlin, M., Buchmann, J.A., Manulis, M., eds.: Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings. Volume 7293 of Lecture Notes in Computer Science., Springer (2012) 190–205
2. Burger, R., Heinle, A.: A diffie-hellman-like key exchange protocol based on multivariate ore polynomials. CoRR **abs/1407.1270** (2014)
3. Matsumoto, T., Imai, H.: Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. In: EUROCRYPT. (1988) 419–453
4. Patarin, J.: Cryptoanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt '88. In Coppersmith, D., ed.: CRYPTO. Volume 963 of Lecture Notes in Computer Science., Springer (1995) 248–261

5. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: EUROCRYPT. (1996) 33–48
6. Berlekamp, E.R.: Factoring polynomials over large finite fields. Mathematics of Computation **24** (1970) pp. 713–735
7. Patarin, J., Goubin, L., Courtois, N.: C^*_{-+} and HM: variations around two schemes of t. matsumoto and h. imai. In Ohta, K., Pei, D., eds.: Advances in Cryptology - ASIACRYPT '98, International Conference on the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings. Volume 1514 of Lecture Notes in Computer Science., Springer (1998) 35–49
8. Patarin, J., Courtois, N., Goubin, L.: Quartz, 128-bit long digital signatures. In Naccache, D., ed.: CT-RSA. Volume 2020 of Lecture Notes in Computer Science., Springer (2001) 282–297
9. Ding, J.: A new variant of the matsumoto-imai cryptosystem through perturbation. In Bao, F., Deng, R.H., Zhou, J., eds.: Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004. Volume 2947 of Lecture Notes in Computer Science., Springer (2004) 305–318
10. Smith-Tone, D.: Properties of the discrete differential with cryptographic applications. [106] 1–12
11. Smith-Tone, D.: On the differential security of multivariate public key cryptosystems. [107] 130–142
12. Perlner, R.A., Smith-Tone, D.: A classification of differential invariants for multivariate post-quantum cryptosystems. [108] 165–173
13. Daniels, T., Smith-Tone, D.: Differential properties of the HFE cryptosystem. [109] 59–75
14. Cartor, R., Gipson, R., Smith-Tone, D., Vates, J.: On the differential security of the hfev- signature primitive. [110] 162–181
15. Vates, J., Smith-Tone, D.: Key recovery attack for all parameters of HFE-. [111] 272–288
16. Cartor, R., Smith-Tone, D.: An updated security analysis of PFLASH. [111] 241–254
17. Patarin, J.: The oil and vinegar algorithm for signatures. Presented at the Dagsthul Workshop on Cryptography (1997)
18. Shamir, A., Kipnis, A.: Cryptanalysis of the oil & vinegar signature scheme. CRYPTO 1998. LNCS **1462** (1998) 257–266
19. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. ACNS 2005, LNCS **3531** (2005) 164–175
20. Moh, T.: A public key system with signature and master key function. Communications in Algebra **27(5)** (1999) 2207–2222
21. Keller, O.H.: Ganze cremona-transformationen. Monatshefte für Mathematik und Physik **47** (1939) 299–306
22. Goubin, L., Courtois, N.T.: Cryptanalysis of the ttm cryptosystem. In Okamoto, T., ed.: Advances in Cryptology — ASIACRYPT 2000, Berlin, Heidelberg, Springer Berlin Heidelberg (2000) 44–57
23. Moh, T., Chen, J.M.: On the goubin-courtois attack on ttm. Cryptology ePrint Archive, Report 2001/072 (2011) https://eprint.iacr.org/2001/072.
24. Chen, J.M., Yang, B.Y., Peng, B.Y.: Tame transformation signatures and topsy-turvy hashes. IWAP (2002) 93–100

25. Ding, J., Schmidt, D.: A defect of the implementation schemes of the TTM cryptosystem. IACR Cryptology ePrint Archive **2003** (2003) 85
26. Moh, T., Chen, J., Yang, B.: Building instances of TTM immune to the goubin-courtois attack and the ding-schmidt attack. IACR Cryptology ePrint Archive **2004** (2004) 168
27. Nie, X., Hu, L., Li, J., Updegrove, C., Ding, J.: Breaking a new instance of TTM cryptosystems. In Zhou, J., Yung, M., Bao, F., eds.: Applied Cryptography and Network Security, 4th International Conference, ACNS 2006, Singapore, June 6-9, 2006, Proceedings. Volume 3989 of Lecture Notes in Computer Science. (2006) 210–225
28. Moh, T.: The recent attack of nie et al on TTM is faulty. IACR Cryptology ePrint Archive **2006** (2006) 417
29. Moh, T.: Two new examples of TTM. IACR Cryptology ePrint Archive **2007** (2007) 144
30. Nie, X., Jiang, X., Hu, L., Ding, J.: Cryptanalysis of two new instances of TTM cryptosystem. IACR Cryptology ePrint Archive **2007** (2007) 381
31. Shamir, A.: Efficient signature schemes based on birational permutations. In Stinson, D.R., ed.: Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings. Volume 773 of Lecture Notes in Computer Science., Springer (1993) 1–12
32. Tsujii, S., Itoh, T., Fujioka, A., Kurosawa, K., Matsumoto, T.: A public-key cryptosystem based on the difficulty of solving a system of nonlinear equations. Systems and Computers in Japan **19** (1988) 10–18
33. Kasahara, M., Sakai, R.: A construction of public-key cryptosystem based on singular simultaneous equations. IEICE Transactions **88-A** (2005) 74–80
34. Wolf, C., Braeken, A., Preneel, B.: Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC. In Blundo, C., Cimato, S., eds.: Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers. Volume 3352 of Lecture Notes in Computer Science., Springer (2004) 294–309
35. Tsujii, S., Gotaishi, M., Tadaki, K., Fujita, R.: Proposal of a signature scheme based on sts trapdoor. [106] 201–217
36. Tadaki, K., Tsujii, S.: Two-sided multiplications are reduced to one-sided multiplication in linear piece in hand matrix methods. In: Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2010, 17-20 October 2010, Taichung, Taiwan, IEEE (2010) 900–904
37. Gotaishi, M., Tsujii, S.: Hidden pair of bijection signature scheme. Cryptology ePrint Archive, Report 2011/353 (2011) http://eprint.iacr.org/.
38. Patarin, J., Goubin, L., Courtois, N.: Improved algorithms for isomorphisms of polynomials. In Nyberg, K., ed.: Advances in Cryptology - EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31 - June 4, 1998, Proceeding. Volume 1403 of Lecture Notes in Computer Science., Springer (1998) 184–200
39. Kipnis, A., Shamir, A.: Cryptanalysis of the hfe public key cryptosystem by relinearization. Advances in Cryptology - CRYPTO 1999, Springer **1666** (1999) 788
40. Courtois, N., Klimov, A., Patarin, J., A.Shamir: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. EUROCRYPT 2000, LNCS **1807** (2000) 392–407

41. Buchberger, B.: A theoretical basis for the reduction of polynomials to canonical forms. SIGSAM Bull. **10** (1976) 19–29
42. Faugere, J.C.: A new efficient algorithm for computing grobner bases (f4). Journal of Pure and Applied Algebra **139** (1999) 61–88
43. Faugere, J.C.: A new efficient algorithm for computing grobner bases without reduction to zero (f5). ISSAC 2002, ACM Press (2002) 75–83
44. Faugere, J.C.: Algebraic cryptanalysis of hidden field equations (hfe) using grobner bases. CRYPTO 2003, LNCS **2729** (2003) 44–60
45. Bouillaguet, C., Cheng, C., Chou, T., Niederhagen, R., Yang, B.: Fast exhaustive search for quadratic systems in $$\mathbb {F}_{2}$$ on fpgas. In Lange, T., Lauter, K.E., Lisonek, P., eds.: Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers. Volume 8282 of Lecture Notes in Computer Science., Springer (2013) 205–222
46. Joux, A., Vitse, V.: A crossbred algorithm for solving boolean polynomial systems. In Kaczorowski, J., Pieprzyk, J., Pomykala, J., eds.: Number-Theoretic Methods in Cryptology - First International Conference, NuTMiC 2017, Warsaw, Poland, September 11-13, 2017, Revised Selected Papers. Volume 10737 of Lecture Notes in Computer Science., Springer (2017) 3–21
47. Niederhagen, R., Ning, K., Yang, B.: Implementing joux-vitse's crossbred algorithm for solving $\mathcal{MQ}$ systems over $\mathbb{F}_2$ on gpus. [112] 121–141
48. Faugère, J., Din, M.S.E., Spaenlehauer, P.: Gröbner bases of bihomogeneous ideals generated by polynomials of bidegree (1, 1): Algorithms and complexity. J. Symb. Comput. **46** (2011) 406–437
49. Berthomieu, J., Faugère, J., Perret, L.: Polynomial-time algorithms for quadratic isomorphism of polynomials. CoRR **abs/1307.4974** (2013)
50. Patarin, J.: Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt '88. Crypto 1995, Springer **963** (1995) 248–261
51. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. EUROCRYPT 1999. LNCS **1592** (1999) 206–222
52. Dubois, V., Fouque, P.A., Shamir, A., Stern, J.: Practical Cryptanalysis of SFLASH. In Menezes, A., ed.: CRYPTO. Volume 4622 of Lecture Notes in Computer Science., Springer (2007) 1–12
53. Patarin, J., Courtois, N., Goubin, L.: Flash, a fast multivariate signature algorithm. CT-RSA 2001, LNCS **2020** (2001) 297–307
54. Fouque, P.A., Macario-Rat, G., Perret, L., Stern, J.: Total break of the ℓic-signature scheme. PKC 2008, LNCS **4939** (2008) 1–17
55. Ding, J., Wolf, C., Yang, B.Y.: l-invertible cycles for multivariate quadratic public key cryptography. PKC 2007 of LNCS **4450** (2007) 266–281
56. Clough, C., Baena, J., Ding, J., Yang, B.Y., Chen, M.S.: Square, a new multivariate encryption scheme. CT-RSA 2009, LNCS **5473** (2009) 252–264
57. Ding, J., Dubois, V., Yang, B.Y., Chen, C.H.O., Cheng, C.M.: Could SFLASH be Repaired? In Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I., eds.: ICALP (2). Volume 5126 of Lecture Notes in Computer Science., Springer (2008) 691–701
58. Moody, D., Perlner, R.A., Smith-Tone, D.: An asymptotically optimal structural attack on the ABC multivariate encryption scheme. [109] 180–196
59. Tao, C., Diene, A., Tang, S., Ding, J.: Simple matrix scheme for encryption. [108] 231–242
60. Moody, D., Perlner, R.A., Smith-Tone, D. In: Key Recovery Attack on the Cubic ABC Simple Matrix Multivariate Encryption Scheme. Springer (2017)

61. Moody, D., Perlner, R.A., Smith-Tone, D.: Improved attacks for characteristic-2 parameters of the cubic ABC simple matrix encryption scheme. [111] 255–271
62. Faugère, J., Gligoroski, D., Perret, L., Samardjiska, S., Thomae, E.: A polynomial-time key-recovery attack on MQQ cryptosystems. In Katz, J., ed.: Public-Key Cryptography - PKC 2015 - 18th IACR International Conference on Practice and Theory in Public-Key Cryptography, Gaithersburg, MD, USA, March 30 - April 1, 2015, Proceedings. Volume 9020 of Lecture Notes in Computer Science., Springer (2015) 150–174
63. Yang, B., Chen, J.: Building secure tame-like multivariate public-key cryptosystems: The new TTS. In Boyd, C., Nieto, J.M.G., eds.: Information Security and Privacy, 10th Australasian Conference, ACISP 2005, Brisbane, Australia, July 4-6, 2005, Proceedings. Volume 3574 of Lecture Notes in Computer Science., Springer (2005) 518–531
64. Porras, J., Baena, J., Ding, J.: Zhfe, a new multivariate public key encryption scheme. [109] 229–245
65. Daniels, T., Smith-Tone, D.: Differential properties of the HFE cryptosystem. IACR Cryptology ePrint Archive **2014** (2014) 398
66. Perlner, R.A., Smith-Tone, D.: Security analysis and key modification for ZHFE. [110] 197–212
67. Chen, M.S., Yang, B.Y., Smith-Tone, D.: Pflash - secure asymmetric signatures on smart cards. Lightweight Cryptography Workshop 2015 (2015) http://csrc.nist.gov/groups/ST/lwc-workshop2015/papers/session3-smith-tone-paper.pdf.
68. Ding, J., Yang, B., Chen, C.O., Chen, M., Cheng, C.: New differential-algebraic attacks and reparametrization of rainbow. In Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M., eds.: Applied Cryptography and Network Security, 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings. Volume 5037 of Lecture Notes in Computer Science. (2008) 242–257
69. Buss, J.F., Frandsen, G.S., Shallit, J.: The computational complexity of some problems of linear algebra. J. Comput. Syst. Sci. **58** (1999) 572–596
70. Bettale, L., Faugère, J., Perret, L.: Cryptanalysis of hfe, multi-hfe and variants for odd and even characteristic. Des. Codes Cryptography **69** (2013) 1–52
71. Faugère, J., Din, M.S.E., Spaenlehauer, P.: Computing loci of rank defects of linear matrices using gröbner bases and applications to cryptology. In Koepf, W., ed.: Symbolic and Algebraic Computation, International Symposium, ISSAC 2010, Munich, Germany, July 25-28, 2010, Proceedings, ACM (2010) 257–264
72. Fröberg, R.: An inequality for Hilbert series of graded algebras. Math. Scand. **56** (1985) 117–144
73. Fouque, P., Granboulan, L., Stern, J.: Differential cryptanalysis for multivariate schemes. In Cramer, R., ed.: Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings. Volume 3494 of Lecture Notes in Computer Science., Springer (2005) 341–353
74. Smith-Tone, D.: The TriTon transformation. Third Workshop on Mathematical Cryptology, WMC 2012. Proceedings, Universidad de Cantabria (2012)
75. Ding, J., Perlner, R.A., Petzoldt, A., Smith-Tone, D.: Improved cryptanalysis of hfev- via projection. [112] 375–395
76. Grover, L.K.: A Fast quantum mechanical algorithm for database search. (1996) Proceedings STOC 1996, 212-219.
77. Bernstein, D.J., Yang, B.: Asymptotically faster quantum algorithms to solve multivariate quadratic equations. [112] 487–506

78. Petzoldt, A., Chen, M., Yang, B., Tao, C., Ding, J.: Design principles for hfev-based multivariate signature schemes. In Iwata, T., Cheon, J.H., eds.: Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I. Volume 9452 of Lecture Notes in Computer Science., Springer (2015) 311–334
79. Dubois, V., Gama, N.: The degree of regularity of HFE systems. In Abe, M., ed.: Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings. Volume 6477 of Lecture Notes in Computer Science., Springer (2010) 557–576
80. Ding, J., Kleinjung, T.: Degree of regularity for HFE-. IACR Cryptology ePrint Archive **2011** (2011) 570
81. Ding, J., Yang, B.Y.: Degree of regularity for hfev and hfev-. [108] 52–66
82. Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y.: Gui. NIST CSRC (2017) https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/Gui.zip.
83. Courtois, N.T.: Generic attacks and the security of quartz. In Desmedt, Y.G., ed.: Public Key Cryptography — PKC 2003, Berlin, Heidelberg, Springer Berlin Heidelberg (2002) 351–364
84. Patarin, J.: La cryptographie multivariable. Mémoire d'habilitation à diriger des recherches de l'Université Paris 7 (1999)
85. Sakumoto, K., Shirai, T., Hiwatari, H.: On provable security of UOV and HFE signature schemes against chosen-message attack. [107] 68–82
86. Cantor, D.G., Zassenhaus, H.: A new algorithm for factoring polynomials over finite fields. Mathematics of Computation **36** (1981) 587–592
87. Cryptographic Technology Group: Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. NIST CSRC (2016) http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf.
88. Casanova, A., Faugere, J.C., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J.: Gemss: A great multivariate short signature. NIST CSRC (2017) https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/GeMSS.zip.
89. Shim, K.A., Park, C.M., Kim, A.: Himq-3: A high-speed signature scheme based on multivariate quadratic equations. NIST CSRC (2017) https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/HiMQ_3.zip.
90. Shim, K., Park, C., Koo, N.: An existential unforgeable signature scheme based on multivariate quadratic equations. In Takagi, T., Peyrin, T., eds.: Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I. Volume 10624 of Lecture Notes in Computer Science., Springer (2017) 37–64
91. Beullens, W., Preneel, B., Szepieniec, A., Vercauteren, F.: Luov signature scheme proposal for nist pqc project. NIST CSRC (2017) https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/LUOV.zip.
92. Beullens, W., Preneel, B.: Field lifting for smaller UOV public keys. In Patra, A., Smart, N.P., eds.: Progress in Cryptology - INDOCRYPT 2017 - 18th International Conference on Cryptology in India, Chennai, India, December 10-13,

2017, Proceedings. Volume 10698 of Lecture Notes in Computer Science., Springer (2017) 227–246
93. 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. Volume 6498 of Lecture Notes in Computer Science., Springer (2010) 33–48
94. Ding, J., Chen, M.S., Petzoldt, A., Schmidt, D., Yang, B.Y.: Rainbow. NIST CSRC (2017) https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/Rainbow.zip.
95. Chen, C.O., Chen, M., Ding, J., Werner, F., Yang, B.: Odd-char multivariate hidden field equations. IACR Cryptology ePrint Archive **2008** (2008) 543
96. Chen, M., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass *MQ* -based identification to *MQ* -based signatures. In Cheon, J.H., Takagi, T., eds.: Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II. Volume 10032 of Lecture Notes in Computer Science. (2016) 135–165
97. Sakumoto, K., Shirai, T., Hiwatari, H.: Public-key identification schemes based on multivariate quadratic polynomials. [113] 706–723
98. Samardjiska, S., Chen, M.S., Hülsing, A., Rijneveld, J., Schwabe, P.: Mqdss specifications. NIST CSRC (2017) https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/MQDSS.zip.
99. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In Buhler, J., ed.: Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings. Volume 1423 of Lecture Notes in Computer Science., Springer (1998) 267–288
100. Bardet, M., Faugére, J., Salvy, B., Yang, B.: Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. Proc. of MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry (2005)
101. Ding, J., Hodges, T.J.: Inverting HFE systems is quasi-polynomial for all fields. [113] 724–742
102. Petzoldt, A.: On the complexity of the hybrid approach on hfev-. IACR Cryptology ePrint Archive **2017** (2017) 1135
103. Petzoldt, A., Bulygin, S., Buchmann, J.A.: Selecting parameters for the rainbow signature scheme. [106] 218–240
104. Faugère, J., Perret, L.: On the security of UOV. IACR Cryptology ePrint Archive **2009** (2009) 483
105. Billet, O., Gilbert, H.: Cryptanalysis of rainbow. In Prisco, R.D., Yung, M., eds.: Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings. Volume 4116 of Lecture Notes in Computer Science., Springer (2006) 336–347
106. Sendrier, N., ed.: Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings. In Sendrier, N., ed.: PQCrypto. Volume 6061 of Lecture Notes in Computer Science., Springer (2010)
107. Yang, B.Y., ed.: Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. In Yang, B.Y., ed.: PQCrypto. Volume 7071 of Lecture Notes in Computer Science., Springer (2011)

108. Gaborit, P., ed.: Post-Quantum Cryptography - 5th International Workshop, PQCrypto 2013, Limoges, France, June 4-7, 2013. Proceedings. In Gaborit, P., ed.: PQCrypto. Volume 7932 of Lecture Notes in Computer Science., Springer (2013)
109. Mosca, M., ed.: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Volume 8772 of Lecture Notes in Computer Science., Springer (2014)
110. Takagi, T., ed.: Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings. Volume 9606 of Lecture Notes in Computer Science., Springer (2016)
111. Lange, T., Takagi, T., eds.: Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings. Volume 10346 of Lecture Notes in Computer Science., Springer (2017)
112. Lange, T., Steinwandt, R., eds.: Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings. Volume 10786 of Lecture Notes in Computer Science., Springer (2018)
113. Rogaway, P., ed.: Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Volume 6841 of Lecture Notes in Computer Science., Springer (2011)

Index