Advances in Mathematics of Communications doi:10.3934/amc.2020043

THE SINGULARITY ATTACK TO THE MULTIVARIATE SIGNATURE SCHEME HIMQ-3

Jintai Ding, Zheng Zhang and Joshua Deaton

Department of Mathematical Science
University of Cincinnati, USA

(Communicated by Tsuyoshi Takagi)

Abstract. We present a cryptanalysis of a signature scheme HIMQ-3 due to Kyung-Ah Shim et al [10], which is a submission to National Institute of Standards and Technology (NIST) standardization process of post-quantum cryptosystems in 2017. We will show that inherent to the signing process is a leakage of information of the private key. Using this information one can forge a signature.

1. Background

1.1. Multivariate public key cryptography. In the past several decades, public key cryptosystems have experienced a rapid development in cryptography. Early public key cryptosystems such as RSA and DSA depend on their difficulty from hard classical number theory. However, Peter Shor's [11] polynomial-time integer factorization algorithm proved that some hard number theory problems, such as the Integer Prime Factorization Problem and the Discrete Logarithm Problem, given the use of a quantum computer. This leads to a crisis in cryptography, and new public-key cryptosystems that have the ability to resist quantum computer attacks are urgently needed. Multivariate public key cryptosystems (MPKC) are considered as candidates of public key cryptosystems that have the potential to resist quantum computer attacks. The security of a MPKC depends on the difficulty of solving a system of multivariate polynomials over a finite field, which has been proved as a NP-complete problem.

1.2. MPKC signature scheme. One of the most well known multivariate public key signature schemes is the Oil Vinegar scheme. The idea of Oil Vinegar signature scheme is that a certain set of the variables are never multiplied together with themselves. If the rest of variables are randomly guessed for, the system will become linear and hopefully have a solution for the message to be signed. The Oil Vinegar schemes can be classified into three groups: Balanced Oil Vinegar [9] (Patarin 1997), Unbalanced Oil Vinegar [8] (Kipnis et al. 1999) and Rainbow [4], a multilayer signature scheme with unbalanced Oil Vinegar at each layer (Ding and Schmidt 2005). The Balanced Oil Vinegar scheme was broken by Kipnis and Shamir[6] using the method of invariant subspaces. The other two scheme types are still unbroken. The HIMQ-3 signature scheme is a multilayer signature scheme which is similar to rainbow.

2010 *Mathematics Subject Classification:* Primary: 58F15, 58F17; Secondary: 53C35.

Key words and phrases: Multivariate public key cryptography, cryptanalysis, oil vinegar signature scheme, multivariate qudratic equations.

1.3. POST-QUANTUM CRYPTOGRAPHY STANDARDIZATION. National Institute of Standards and Technology (NIST)[5] believes that it is prudent to begin developing standards for post-quantum cryptography because of the fast development of quantum computers. These new standards will be used as quantum resistant counterparts to existing standards. By the end of 2017, 23 signature schemes and 59 encryption/KEM schemes were submitted, of which 69 participated in the first round. HIMQ-3 is a round 1 candidate on the list.

2. HIMQ-3 SIGNATURE SCHEME

2.1. CYCLE PRODUCTS [6]. The HIMQ-3 scheme contains a system of quadratic equations called cycle products. The system makes it possible to invert the central map.

Suppose $\mathbb{F}_q$ is a field of characteristic 2 and l is an odd positive integer. The cycle products system $\mathcal{Q}$ is defined by:

$$\mathcal{Q}: \alpha_1 x_1 x_2 = \beta_1, \alpha_2 x_2 x_3 = \beta_2, \cdots, \alpha_l x_l x_1 = \beta_l,$$

where α_i and β_i are nonzero elements in $\mathbb{F}_q$.

To find a solution to $\mathcal{Q}$, first write the cycle products in the form

$$x_1 x_2 = \gamma_1, \cdots, x_l x_1 = \gamma_l,$$

where $\gamma_i = \beta_i/\alpha_i$. Let $A = \gamma_1\gamma_2\cdots\gamma_l$ and $B = \gamma_2\gamma_4\cdots\gamma_{l-1}$. It is easy to see that $x_1 = \sqrt{A}/B$, $x_i = \gamma_{i-1}/x_{i-1}$ for $i = 2, \cdots, l-1$, and $x_l = \gamma_l/x_1$. Critical is the observation that this means for any given cycle product of the form above, a given solution will never contain zero. This is the crux of our attack.

2.2. HIMQ-3 SCHEME [10]. Let us denote $\mathbb{F}_q$ to be the finite field of order $q = 2^k$. Let v, o_1, o_2, o_3 be positive integers where o_1 and o_2 are odd. Further, let $v_1 = v+o_1$, $v_2 = v+o_1+o_2$, $m = o_1+o_2+o_3$ and $n = v+o_1+o_2+o_3$. Let $\mathbf{X} = (x_1, \cdots, x_n)$. Let $\mathcal{F} = (\mathcal{F}^{(1)}, \cdots, \mathcal{F}^{(m)})$ be the central map defined by three layers:

Layer 1: For $i = 1, \cdots, o_1-1$, $\mathcal{F}^{(i)}(\mathbf{X}) = \Phi_i(\mathbf{X}) + \delta_i x_{v+i} x_{v+i+1}$, and for $i = o_1$, $\mathcal{F}^{(o_1)}(\mathbf{X}) = \Phi_{o_1}(\mathbf{X}) + \delta_{o_1} x_{v+o_1} x_{v+1}$; where $\Phi_i(\mathbf{X}) = \sum_{j=1}^{v} \alpha_{i,j} x_j x_{1+(i+j-1)(\mathrm{mod}\ v)}$ with $\alpha_{i,j}$ a nonzero element in $\mathbb{F}_q$.

Layer 2: For $i = 1, \cdots, o_2-1$, $\mathcal{F}^{(o_1+i)}(\mathbf{X}) = \Psi_i(\mathbf{X}) + \delta_{o_1+i} x_{v_1+i} x_{v_1+i+1}$, and for $i = o_2$, $\mathcal{F}^{(o_1+o_2)}(\mathbf{X}) = \Psi_{o_2}(\mathbf{X}) + \delta_{o_1+o_2} x_{v_1+o_2} x_{v_1+1}$; where $\Psi_i(\mathbf{X})$ is a quadratic polynomial in the variables $(x_1, \cdots, x_{v+o_1})$ defined by $\Psi_i(\mathbf{X}) = \sum_{j=1}^{v} \alpha'_{i,j} x_j x_{v+(i+j-1)(\mathrm{mod}\ o_1)}$ with $\alpha'_{i,j}$ a nonzero element in $\mathbb{F}_q$.

Layer 3: For $i = 1, \cdots, o_3$, $\mathcal{F}^{(o_1+o_2+i)}(\mathbf{X}) = \sum_{v+1 \le l \le j \le v_1} \beta^{(i)}_{l,j} x_l x_j + \Theta_i(\mathbf{X}) + \Theta'_i(\mathbf{X}) + \epsilon_i x_{o_1+o_2+i}$; where $\beta^{(i)}_{l,j} \in \mathbb{F}_q$, and Θ_i and Θ'_i are quadratics in variables $(x_1, \cdots, x_n)$ defined by $\Theta_i(\mathbf{X}) = \sum_{j=1}^{v_1} \gamma_{i,j} x_j x_{v_1+(i+j-1)(\mathrm{mod}\ o_2)}$, $\Theta'_i(\mathbf{X}) = \sum_{j=1}^{v_2} \gamma'_{i,j} x_j x_{v_2+(i+j-1)(\mathrm{mod}\ o_3)}$ with $\gamma_{i,j}$ and $\gamma'_{i,j}$ nonzero elements in $\mathcal{F}_q$.

To hide the ability to find pre-images and thus construct a public key from $\mathcal{F}$, one uses two invertible affine maps $\mathcal{S}: \mathbb{F}_q^m \to \mathbb{F}_q^m$, and $\mathcal{T}: \mathbb{F}_q^n \to \mathbb{F}_q^n$. The public key is the composition $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$. The private keys are the invertible affine maps $\mathcal{S}$ and $\mathcal{T}$. The signing process for a document is as follows:

$$\mathbb{F}_q^m \xrightarrow{\mathcal{S}^{-1}} \mathbb{F}_q^m \xrightarrow{\mathcal{F}^{-1}} \mathbb{F}_q^n \xrightarrow{\mathcal{T}^{-1}} \mathbb{F}_q^n$$

The verification process is just backwards

$$\mathbb{F}_q^m \xleftarrow{\mathcal{P}} \mathbb{F}_q^n$$

2.3. Inverting the central map. Given a $M = (M_1, \cdots, M_m)$ in $\mathbb{F}_q^m$, we want to compute $\mathcal{F}^{-1}(M) = \mathbf{s}$.
1. Randomly generate $\mathbf{s_v} \in \mathbb{F}_q^v$, and plug $\mathbf{s_v}$ into the first layer obtaining the cycle product

$$\begin{cases} \delta_1 x_{v+1} x_{v+2} = M_1 - \Phi_1(\mathbf{s_v}) \\ \vdots \\ \delta_{o_1} x_{v+o_1} x_{v+1} = M_{o_1} - \Phi_{o_1}(\mathbf{s_v}). \end{cases}$$

2. If $M_i - \Phi_i(\mathbf{s_v}) \neq 0$ for all i, then solve by the process described before. Name this $\mathbf{s_{v_1}} \in \mathbb{F}_q^{v_1}$. Otherwise, return to step 1.
3. Plug $\mathbf{s_{v_1}}$ into the second layer creating another cycle product

$$\begin{cases} \delta_{o_1+1} x_{v_1+1} x_{v_1+2} = M_{o_1+1} - \Psi_1(\mathbf{s_{v_1}}) \\ \vdots \\ \delta_{o_1+o_2} x_{v_1+o_2} x_{v_1+1} = M_{o_1+o_2} - \Psi_{o_1}(\mathbf{s_{v_1}}). \end{cases}$$

If $M_{o_1+i} - \Psi_i(\mathbf{s_v}) \neq 0$ for all i, call the solution $\mathbf{s_{v_2}} \in \mathbb{F}_q^{v_2}$. Otherwise, return to step 1.
4. Plug $\mathbf{s_{v_2}}$ into the third layer. It will thus have only linear terms. Use Gaussian Elimination to see if there is a solution. If so, then the solution is $\mathbf{s}$. Otherwise, return to step 1.

3. The singularity attack

3.1. Key observation. The weakness of HIMQ-3 is that the cycle variables cannot be equal to zero when evaluated at a honestly generated signature. In addition, this fact does not change under a change of basis $\mathcal{T}$. Since the scheme is constructed over a finite field of 2^k elements, it is a basic knowledge that if we raise any nonzero element a in the field to the power of $2^k - 1$, then $a^{2^k-1} = 1$. For this reason, if we evaluate the cycle variables at signatures under the action of $\mathcal{T}$, and then raise the power, we will obtain some equations. Once we solve these equations, we will get part of the private key up to unit multiplication.

3.2. Finding parts of $\mathcal{T}$. Suppose that a private key $(\mathcal{F}, \mathcal{T}, \mathcal{S})$ has been generated with its corresponding public key $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$. We may describe the affine map $\mathcal{T}$ by an invertible matrix $(a_{ij})_{1 \leq i,j \leq n}$ and a vector $b = (b_1, \cdots, b_n)$ so that for any $(x_1, \cdots, x_n) \in \mathbb{F}_q^n$ we have that

$$\mathcal{T}((x_1, \cdots, x_n)) = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n a_{1i} x_i + b_1 \\ \sum_{i=1}^n a_{2i} x_i + b_2 \\ \vdots \\ \sum_{i=1}^n a_{ni} x_i + b_n \end{bmatrix}.$$

Our goal is to find how $\mathcal{T}$ mixes the variables used in the cycle products up to a multiplication by a non-zero constant. In other words, we only need solutions for a_{ji} and b_j up to unit multiplication. If we have a signature $\sigma = (\sigma_1, \cdots, \sigma_n)$, then

for $v+1 \leq j \leq v+o_1+o_2$, we have $\sum_{i=1}^{n} a_{ji}\sigma_i + b_j \neq 0$. This allows us to say that for any $\gamma_j \in \mathbb{F}_q^*$ and signature σ

$$1 = \left(\sum_{i=1}^{n} \gamma_j a_{ji}\sigma_i + \gamma_j b_j\right)^{2^k-1} = \prod_{h=1}^{k}\left(\sum_{i=1}^{n} \gamma_j a_{ji}\sigma_i + \gamma_j b_j\right)^{2^{k-h}}$$
$$= \prod_{h=1}^{k}\left(\sum_{i=1}^{n} (\gamma_j a_{ji}\sigma_i)^{2^{k-h}} + (\gamma_j b_j)^{2^{k-h}}\right).$$

First we will solve the case when $b_j \neq 0$, the case for $b_j = 0$ is essentially the same. Let $\gamma_j = b_j^{-1}$, we obtain

$$\prod_{h=1}^{k}\left(\sum_{i=1}^{n} (b_j^{-1}a_{ji}\sigma_i)^{2^{k-h}} + 1\right) = 1.$$

For the sake of notation let $\tilde{a}_{ji} = b_j^{-1}a_{ij}$. Thus we see by performing the above product that

$$\tilde{a}_{j1}^{2^k-1}\sigma_1^{2^k-1} + \tilde{a}_{j1}^{2^k-2}\tilde{a}_{j2}\sigma_1^{2^k-2}\sigma_2 + \cdots + \tilde{a}_{jn}\sigma_n + 1 = 1.$$

We can treat the individual products of the $\tilde{a}_{ij}$'s as individual variables to get a homogeneous linear equation with $(n+1)^k - 1$ terms. We get another homogeneous linear equation if we use a different signature. Hence by collecting $(n+1)^k - 1$ signatures we can form a matrix in the following way.

3.2.1. *Construction of the matrix.* For $v+1 \leq j \leq v+o_1+o_2$, we list the products of $\tilde{a}_{ij}$ in the order: $\tilde{a}_{j1}^{2^k-1}, \tilde{a}_{j1}^{2^k-2}\tilde{a}_{j2}, \cdots, \tilde{a}_{jn}$ (Here we use lexicographic order on $(l_1, -l_1', l_2, -l_2', ...)$ for products $\tilde{a}_{jl_1}^{l_1'}\tilde{a}_{jl_2}^{l_2'}...$). Moreover, for each signature $\sigma = (\sigma_1, \cdots, \sigma_n)$, the corresponding coefficients are: $\sigma_1^{2^k-1}, \sigma_1^{2^k-2}\sigma_2, \cdots, \sigma_n$. The matrix is simply constructed by having these corresponding coefficients as a row for each signature we use. Therefore the size of this matrix is $(n+1)^k-1$ by $(n+1)^k-1$ if we use $(n+1)^k - 1$ signatures. Hence, we obtain a homogeneous linear system: $\mathbf{Ax} = \mathbf{0}$, where $\mathbf{A}$ is the matrix whose rows are $(\sigma_1^{2^k-1}, \sigma_1^{2^k-2}\sigma_2, \cdots, \sigma_n)$ for each signature used, and $\mathbf{x} = (\tilde{a}_{j1}^{2^k-1}, \tilde{a}_{j1}^{2^k-2}\tilde{a}_{j2}, \cdots, \tilde{a}_{jn})^T$.

Remark 1. Assume that $b_j \neq 0$, for $v+1 \leq j \leq v+o_1+o_2$, $\tilde{\mathbf{a}}_j = (\tilde{a}_{j1}^{2^k-1}, \tilde{a}_{j1}^{2^k-2}\tilde{a}_{j2}, \cdots, \tilde{a}_{jn})^T$ is contained in the kernel of $\mathbf{A}$. Moreover, it is obvious that they are linearly independent. It follows that $\mathrm{Rank}(\mathbf{A}) \leq (n+1)^k - 1 - (o_1+o_2)$. In fact, according to our experiments (see chapter 4), with very high probability, $\mathrm{Rank}(\mathbf{A}) = (n+1)^k - 1 - (o_1+o_2)$.

3.2.2. *Solving the $\tilde{a}_{ji}$'s.* We do Gaussian Elimination on this matrix $\mathbf{A}$, and turn the linear system into $\mathbf{A'x} = \mathbf{0}$. Now we try to solve for the $\tilde{a}_{ji}$'s. We start at the bottom of $\mathbf{A'}$. If $\mathbf{A}$ has rank $(n+1)^k - 1 - (o_1+o_2)$, then in the last nonzero row of $\mathbf{A'}$, most entries will equal to zero and the nonzero entries will only appear in the last o_1+o_2+1 columns in variables $\tilde{a}_{jn}^{o_1+o_2+1}, \tilde{a}_{jn}^{o_1+o_2}, \tilde{a}_{jn}^{o_1+o_2-1}, \cdots, \tilde{a}_{jn}$. Hence, converting this back into a polynomial means we have a univariate polynomial equation which we can thus solve. One can see that if $2^k - 1 \geq o_1+o_2+1$, we will obtain a univeriate polynomial. This allows us to get our possibilities for $\tilde{a}_{jn}$ (as the above equation will be true for any of the $\tilde{a}_{ji}$'s, $v+1 \leq j \leq v+o_1+o_2$, we will return all of these values). We then move up the matrix to the first time

that $\tilde{a}_{j(n-1)}$ appears only with powers of itself and $\tilde{a}_{jn}$. As we already know what $\tilde{a}_{jn}$ can be, this is also a univariate polynomial equation. For each of our possible solutions to $\tilde{a}_{jn}$, we plug in and get the possible solutions to $\tilde{a}_{j(n-1)}$. Continue this process until we collect all the $\tilde{a}_{ji}$ for which $b_j \neq 0$. On the other hand, to avoid the inequality $2^k - 1 \geq o_1 + o_2 + 1$, the size of the field is then forced to be small, which reduces the complexity of other attacks such as direct attack, min/high rank attack (see Section 2.2, 2.3 and 2.4 in [10]). The process is essentially the same as for the case $b_j = 0$ except that we then guess the last available $\tilde{a}_{ji}$ to be non-zero hence enabling us to set $\gamma_j = \tilde{a}_{ji}^{-1}$ for that particular $\tilde{a}_{ji}$. Repeat until all of the $\tilde{a}_{ji}$ are found, which generally is after the first few guesses. Note that the collection of $\tilde{a_{ji}}$ that we found can recover the cycle variables. A toy example is provided in the appendix.

3.3. Getting public key into right form. The second step is to convert the public key into an equivalent HIMQ-3 central map. Observe that second layer polynomials will vanish but not those from the first and third if we set the cycle variables to be zero. So we can kill the second layer by setting cycle variables equal to zero, then apply the Gaussian Elimination to separate the second layer. In addition we want to remove the o_3 variables from all but the third layer, and this can be done given that the image of the o_3 variables lies in the kernel of symmetric matrices of second layer. To separate first and second layer. This can be achieved by using linear combinations of the symmetric matrices to reduce the rank. The third layer is not of importance because it is essentially an oil Vinegar layer, and the values for the vinegar variables will be found using the first two layers. The change of basis transformation is formed by the images of v, o_1, o_2, and o_3 variables, which we get using the images and kernels of the symmetric matrices. After applying change of basis, we can see that the matrices of the first and second layer are nearly in the form that we want besides for some slight indices problem arising from us not knowing the order of the variables. This is easily fixed.

4. Complexity and implement

In our attack, the most complicated step is to do Gaussian elimination over a linear system of dimension $(n+1)^k - 1$. The complexity of solving such linear system is $((n+1)^k - 1)^\omega$, where ω called the complexity exponent of linear algebra [1] and it depends on the algorithm we choose. The best published estimates to date gives $\omega \approx 2.3727$ [13][7]. In addition many people believe that the true value of ω is 2 [13][3][2]. A practical algorithm that is frequently used for implementation is Strassen-Winograd's algorithm [12] with $\omega \approx 2.8047$. For $v = 31$, $o_1 = o_2 = 15, o_3 = 14$ and $k = 8$, the parameters for 128-bit security parameter proposed in [10], we need approximately 2^{50} signatures, and we estimate the complexity to be from 2^{119} using [13] and 2^{140} using [12].

We ran our experiments with magma of version V2 24-10 on 3.6 GHz Intel Core i7 and 8GB of 2666 MHz DDR4 RAM. We attacked the scheme with two sets of parameters. For $v = 7, o_1 = 3, o_2 = 3, o_3 = 2, k = 3$, in 1000 attempts, the rank of the matrix is always $(n+1)^k - 1 - (o_1 + o_2) = 4089$, and we can always get parts of $\mathcal{T}$. For $v = 9, o_1 = o_2 = 3, o_3 = 2, k = 3$, in 1000 attempts, the rank of the matrix is always $(n+1)^k - 1 - (o_1 + o_2) = 5825$, and we can always get parts of $\mathcal{T}$.

APPENDIX A. APPENDIX: TOY EXAMPLE

We provide a toy example to clarify the step **3.2**. In this example, we choose $k = 3$, thus our field is the Galois field of 2^3 elements. The Galois field will be represented by $\{0, 1, w, w^2, \cdots, w^6\}$, where w is a generator in the multiplicative group of the Galois field. Let $n = 2$. For the sake of clarity. We use a linear map instead of a affine map. Our linear map $\mathcal{T}$ is randomly chosen to be the matrix $\begin{bmatrix} w^2 & w^2 \\ w^3 & w \end{bmatrix}$.

Suppose we obtain a set of signatures (x_1, x_2):

$$(w, w^5), (w^5, w), (w^2, 1), (w^6, w^5), (0, w^2), (w^5, w^3), (1, w^6), (0, w^5),$$
$$(0, w^2), (1, 0), (w^5, w^6), (0, w), (w^5, w^3), (1, w), (w^5, 0), (w^6, 1), (w^6, w^3),$$
$$(w, w^4), (w^2, w^5), (w^3, w), (1, w^6), (w, 1), (w^2, w), (w^2, w), (w^4, w), (w^4, 1), (w^4, w^2).$$

We first construct a generic polynomial $g = a_1x_1 + a_2x_2$. We assume that this polynomial is never equal to zero. Hence, in this Galois field, $g^{2^3-1} = (a_1x_1 + a_2x_2)^{2^3-1} = 1$. By elementary field theory, we can rewrite this equation as

$$(a_1x_1 + a_2x_2)^{2^3-1} = (a_1x_1 + a_2x_2)^{2^{3-1}}(a_1x_1 + a_2x_2)^{2^{3-2}}(a_1x_1 + a_2x_2)^{2^{3-3}} = 1$$

Since this is a field of characteristic 2, the equations turns out to be

$$((a_1x_1)^{2^{3-1}} + (a_2x_2)^{2^{3-1}})((a_1x_1)^{2^{3-2}} + (a_2x_2)^{2^{3-2}})((a_1x_1)^{2^{3-3}} + (a_2x_2)^{2^{3-3}}) = 1$$

Multiply the product out, we have

$$a_1^7x_1^7 + a_1^6a_2x_1^6x_2 + a_1^5a_2^2x_1^5x_2^2 + a_1^4a_2^3x_1^4x_2^3 + a_1^3a_2^4x_1^3x_2^4 +$$

$$a_1^2a_2^5x_1^2x_2^5 + a_1a_2^6x_1x_2^6 + a_2^7x_2^7 + 1 = 0$$

We view the products of a_i as variables, and x_i as coefficients. In other words, we have the coefficients in the order:

$$x_1^7, x_1^6x_2, x_1^5x_2^2, x_1^4x_2^3, x_1^3x_2^4, x_1^2x_2^5, x_1x_2^6, x_2^7, 1$$

and monomials in the order:

$$a_1^7, a_1^6a_2, a_1^5a_2^2, a_1^4a_2^3, a_1^3a_2^4, a_1^2a_2^5, a_1a_2^6, a_2^7, 1$$

If we evaluate these coefficients at the signatures, we get $(n+1)^k$ vectors which will be the rows of the following matrix:

$$\begin{bmatrix}
1 & w^4 & w & w^5 & w^2 & w^6 & w^3 & 1 & 1 \\
1 & w^3 & w^6 & w^2 & w^5 & w & w^4 & 1 & 1 \\
1 & w^5 & w^3 & w & w^6 & w^4 & w^2 & 1 & 1 \\
1 & w^6 & w^5 & w^4 & w^3 & w^2 & w & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & w^5 & w^3 & w & w^6 & w^4 & w^2 & 1 & 1 \\
1 & w^6 & w^5 & w^4 & w^3 & w^2 & w & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & w & w^2 & w^3 & w^4 & w^5 & w^6 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & w^5 & w^3 & w & w^6 & w^4 & w^2 & 1 & 1 \\
1 & w & w^2 & w^3 & w^4 & w^5 & w^6 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & w & w^2 & w^3 & w^4 & w^5 & w^6 & 1 & 1 \\
1 & w^4 & w & w^5 & w^2 & w^6 & w^3 & 1 & 1 \\
1 & w^3 & w^6 & w^2 & w^5 & w & w^4 & 1 & 1 \\
1 & w^3 & w^6 & w^2 & w^5 & w & w^4 & 1 & 1 \\
1 & w^5 & w^3 & w & w^6 & w^4 & w^2 & 1 & 1 \\
1 & w^6 & w^5 & w^4 & w^3 & w^2 & w & 1 & 1 \\
1 & w^6 & w^5 & w^4 & w^3 & w^2 & w & 1 & 1 \\
1 & w^6 & w^5 & w^4 & w^3 & w^2 & w & 1 & 1 \\
1 & w^6 & w^5 & w^4 & w^3 & w^2 & w & 1 & 1 \\
1 & w^4 & w & w^5 & w^2 & w^6 & w^3 & 1 & 1 \\
1 & w^3 & w^6 & w^2 & w^5 & w & w^4 & 1 & 1 \\
1 & w^5 & w^3 & w & w^6 & w^4 & w^2 & 1 & 1
\end{bmatrix}$$

We apply echelon form on this matrix and then remove the zero rows. The new matrix is:

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & w^5 & 0 & w^4 \\
0 & 0 & 1 & 0 & 0 & 0 & w^2 & 0 & w^6 \\
0 & 0 & 0 & 1 & 0 & 0 & w^4 & 0 & w^5 \\
0 & 0 & 0 & 0 & 1 & 0 & w^3 & 0 & w \\
0 & 0 & 0 & 0 & 0 & 1 & w^6 & 0 & w^2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}$$

Our next goal is to turn this matrix back to polynomials. Recall the order of the monomials, we get 7 multivariate polynomials:

$$\begin{aligned}
&a_1^7 + 1 \\
&a_1^6 a_2 + w^5 a_1 a_2^6 + w^4 \\
&a_1^5 a_2^2 + w^2 a_1 a_2^6 + w^6 \\
&a_1^4 a_2^3 + w^4 a_1 a_2^6 + w^5 \\
&a_1^3 a_2^4 + w^3 a_1 a_2^6 + w \\
&a_1^2 a_2^5 + w^6 a_1 a_2^6 + w^2 \\
&a_2^7 + 1
\end{aligned}$$

The first and last polynomials do not help, they are trivial. Remember that we are not looking for the original values for a_i, we only need solutions for a_i up to unit multiple. Therefore, we can set $a_1 = 1$, and if we pick the second polynomial, we then get a univariate polynomial $w^5 a_2^6 + a_2 + w^4$. The roots are $a_2 = 1$ and $a_2 = w^5$.

Let us check our solution with the linear map $\mathcal{T} = \begin{bmatrix} w^2 & w^2 \\ w^3 & w \end{bmatrix}$. It is clear that $a_1 = 1$ and $a_2 = 1$ are unit multiples of $a_1 = w^2$ and $a_2 = w^2$. Now if we check the second row, The original values are:

$$a_1 = w^3$$
$$a_2 = w$$

If we multiply the inverse of w^3 by w, we get w^{-2} which is exactly equal to w^5 in the Galois field of 2^3 elements.

REFERENCES

[1] M. Albrecht, G. Bard and C. Pernet, Efficient dense Gaussian elimination over the finite field with two elements, preprint, arXiv:1111.6549.

[2] H. Cohn, R. Kleinberg, B. Szegedy and C. Umans, Group-theoretic algorithms for matrix multiplication, *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, (2005), 379–388.

[3] D. Coppersmith and S. Winograd, Matrix multiplication via arithmetic progressions, *Journal of symbolic computation*, **9** (1990), 251–280.

[4] J. Ding and D. Schmidt, Rainbow, a new multivariable polynomial signature scheme, *International Conference on Applied Cryptography and Network Security Springer*, (2005), 164–175.

[5] National Institute of Standards and Technology, *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process*, 2017. Available from: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf.

[6] J. T. Ding, C. Wolf and B.-Y. Yang, l-invertible cycles for Multivariate Quadratic (MQ) public key cryptography, *Public Key Cryptography-PKC 2007, Lecture Notes in Comput. Sci., Springer, Berlin*, **4450** (2007), 226–281.

[7] J. Dumas and C. Pernet, Computational linear algebra over finite fields, preprint, arXiv:1204.3735.

[8] A. Kipnis, J. Patarin and L. Goubin, Unbalanced oil and vinegar signature schemes, *Advances in Cryptology—EUROCRYPT '99 (Prague), Lecture Notes in Comput. Sci., Springer, Berlin*, **1592** (1999), 206–222.

[9] J. Patarin, The oil and vinegar algorithm for signatures, in *Dagstuhl Workshop on Cryptography*, (1997).

[10] K. Shim, C. Park and A. Kim, Himq-3: A high speed signature scheme based on multivariate quadratic equations, (2017).

[11] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review*, **41** (1999), 303–332.

[12] V. Strassen, Gaussian elimination is not optimal, *Numerische Mathematik*, **13** (1969), 354–356.

[13] V. V. Williams, Breaking the Coppersmith-Winograd barrier, CiteSeer, Available from: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.228.9947&rep=rep1&type=pdf.

Received March 2019; 1st revision July 2019; final revision September 2019.

E-mail address: jintai.ding@gmail.com
E-mail address: zhzhang1989@gmail.com
E-mail address: deatonju@mail.uc.edu