Cryptanalysis of The Lifted Unbalanced Oil Vinegar Signature Scheme

Jintai Ding, Joshua Deaton, Kurt Schmidt, Vishakha, and Zheng Zhang

University of Cincinnati, OH, USA
jintai.ding@gmail.com
{deatonju,schmidku,sharmav4,zhang2zh}@mail.uc.edu

Abstract. In 2017, Ward Beullens *et al.* submitted Lifted Unbalanced Oil and Vinegar (LUOV)[4], a signature scheme based on the famous multivariate public key cryptosystem (MPKC) called Unbalanced Oil and Vinegar (UOV), to NIST for the competition for post-quantum public key scheme standardization. The defining feature of LUOV is that, though the public key $\mathscr{P}$ works in the extension field of degree r of $\mathbb{F}_2$, the coefficients of $\mathscr{P}$ come from $\mathbb{F}_2$. This is done to significantly reduce the size of $\mathscr{P}$. The LUOV scheme is now in the second round of the NIST PQC standardization process.
In this paper we introduce a new attack on LUOV. It exploits the "lifted" structure of LUOV to reduce direct attacks on it to those over a subfield. We show that this reduces the complexity below the targeted security for the NIST post-quantum standardization competition.

1 Introduction

1.1 Background and Post-Quantum Cryptography Standardization

A crucial building block for any free, secure, and *digital* society is the ability to authenticate digital messages. In their seminal 1976 paper [40], Whitfield Diffie and Martin Hellman described the mathematical framework to do such, which is now called a digital signature scheme. They proposed the existence of a function F so that for any given message D any party can easily check whether for any X that $F(X) = D$, *i.e.* verify a signature. However, only one party, who has a secret key, can find such an X, *i.e.* sign a document. Such a function F is called a trapdoor function. Following this idea, Rivest, Shamir, and Adleman proposed the first proof of concept of a signature scheme based on their now famous RSA public key encryption scheme, which relies on the difficulty of integer factorization [38].

Up to 2013, the National Institute of Standards and Technology (NIST)'s guidelines allowed for three different types of signature schemes: the Digital Signature Algorithm (DSA), RSA Digital Signature Algorithm, and The Elliptic Curve Digital Signature Algorithm [25]. However, a major drawback to these signature schemes is that in 1999 Peter Shor showed that they were weak to a sufficiently powerful quantum computer [39]. As research towards developing a fully fledged quantum computer continues, it has become increasingly clear that there is a significant need to prepare our current communication infrastructure for a post-quantum world. For it is not

easy nor quick undergoing to transition our current infrastructure into a post quantum one. Thus, a significant effort will be required in order to develop, standardize, and deploy new post-quantum signature schemes.

As such in December 2016, NIST, under the direction of the NSA, put out a call for proposals of new post-quantum cryptosystems. NIST expects to perform multiple rounds of evaluations over a period of three to five years. The goal of this process is to select a number of acceptable candidate cryptosystems for standardization. These new standards will be used as quantum resistant counterparts to existing standards. The evaluation will be based on the following three criteria: Security, Cost, and Algorithm and Implementation Characteristics. We are currently in the second round of this process, and out of the original twenty-three signature schemes there are only nine left. LUOV is one of these remaining.

An additional complication to designing a post-quantum cryptosystem is quantifying security levels in a post quantum world for the exact capabilities of a quantum computer is not fully understood. In [34], NIST addresses this issue and quantifies the security strength of a given cryptosystem by comparing it to existing NIST standards in symmetric cryptography, which NIST expects to offer significant resistance to quantum cryptanalysis. Below are the relevant NIST security strength categories which we present the log base 2 of the complexity.

NIST Level	Security Description	Complexity
II	At least as hard to break as SHA256 (collision search)	146
IV	At least as hard to break as SHA384 (collision search)	210
V	At least as hard to break as AES256 (exhaustive key search)	272

Table 1. Description of different NIST security strength categories.

1.2 Multivariate Public Key Cryptosystems

Since the work of Diffie and Hellman, mathematicians have found many other groups of cryptosystems that do not rely on Number Theory based problems. Some of these seem to be good candidates for a post-quantum system. One such group is Multivariate Public Key Cryptosystems (MPKC)[12][15]. The security of MPKC depends on the difficulty of solving a system of m multivariate polynomials in n variables over a finite field. Usually these polynomials are of degree two. Solving a set of random multivariate polynomial equations over a finite field is proven to be an NP-hard problem [27], thus lending a solid foundation for a post-quantum signature scheme. Furthermore, MPKCs in general can be computationally much more efficient than many other systems. However, as these systems need to be made into a trapdoor function they cannot be truly random. They must be of a special form, which is generally hidden by composition with invertible linear maps. The difficulty lies in creating a hidden structure which does not impact the difficulty of solving the system.

A breakthrough in MPKC was proposed by Matsumoto and Imai in 1988 which is called either the MI cryptosystem or C^* [30]. They worked with a finite field k, but they did not work with the vector space k^n directly. Instead, they looked to a degree n extension of k where an inverse map can be constructed which is still a trapdoor function. As such this can be used to both encrypt and sign documents . This scheme was broken by Patarin using the Linearization Equation Attack which is the inspiration for all Oil and Vinegar Schemes [35]. To be brief, Patarin discovered that plain-text/cipher-text pairs $(\mathbf{x}, \mathbf{y})$ will satisfy equations (called the linearization equations) of the form

$$\sum \alpha_{ij} x_i y_j + \sum \beta_i x_i + \sum \gamma_i y_i + \delta = 0$$

Collecting enough such pairs and plugging them into above equations produces linear equations in the α_{ij}'s, β_i's, γ_i's, and δ which then can be solved for. Then for any cipher-text $\mathbf{y}$, its corresponding plain-text $\mathbf{x}$ will satisfy the linear equations found by plugging in $\mathbf{y}$ into the linearization equations. This will either solve for the $\mathbf{x}$ directly if enough linear equations were found or at least massively increase the efficiency of other direct attacks of solving for $\mathbf{x}$. Inspired by the attack, Patarin introduced the Oil and Vinegar scheme [36]. This has been one of the most studied schemes for multivariate cryptography.

1.3 A Brief Sketch and History of Oil and Vinegar Schemes

One of the most well known multivariate public key signature schemes is the Oil and Vinegar scheme. The key idea of the Oil and Vinegar signature scheme is to reduce signing a document into solving a linear system. This is done by separating the variables into two collections, the vinegar variables and the oil variables. Let $\mathbb{F}$ be a (generally small) finite field, o and v be two integers, and $n = o + v$. The central map $\mathscr{F} : \mathbb{F}^n \to \mathbb{F}^o$ is a quadratic map whose components $f_1, \ldots, f_o$ are in the form

$$f_k(\mathbf{x}) = \sum_{i=1}^{v} \sum_{j=i}^{n} \alpha_{i,j,k} x_i x_j + \sum_{i=1}^{n} \beta_{i,k} x_i + \gamma_k$$

where each coefficient is in $\mathbb{F}$. Here, $x_1, \ldots, x_v$ (which are called the vinegar variables) are potentially multiplied to all the other variables including themselves. However, the variables $x_{v+1}, \ldots, x_n$ (which are called the oil variables) are never multiplied to one another. Hence, if one guesses for all the vinegar variables, one is left with a system of o linear polynomials in o variables. This has a high probability of being invertible, and if it is not one can just take another guess for the vinegar variables. Hence to find pre-images for $\mathscr{F}$, one repeatedly guesses values for the vinegar variables until the resulting linear system is invertible. The public key $\mathscr{P}$ is the composition of $\mathscr{F}$ with an invertible affine map $\mathscr{T} : \mathbb{F}^n \to \mathbb{F}^n$.

$$\mathscr{P} = \mathscr{F} \circ \mathscr{T}.$$

The private key pair is $(\mathscr{F}, \mathscr{T})$. To find a signature for a message $\mathbf{y}$, one first finds an element z in $\mathscr{F}^{-1}(y)$, and then simply computes a signature by finding $\mathscr{T}^{-1}(z)$.

The security of Oil and Vinegar schemes relies on the fact that $\mathscr{P}$ is essentially as hard to find pre-images for as a random system (when one does not know the decomposition).

Patarin originally proposed that the number of oil variables would equal the number of vinegar variables. Hence the the original scheme is now called Balanced Oil and Vinegar. However, Balanced Oil Vinegar was broken by Kipnis and Shamir using the method of invariant subspaces [28]. This attack, however, is thwarted by making the number of vinegar variables sufficiently greater then the number of oil variables. The other major attack using the structure of UOV is the Oil and Vinegar Reconciliation attack proposed by Ding *et al.* However, with appropriate parameters this attack can be avoided as well [18].

Proposed nearly twenty years ago, the Unbalanced Oil and Vinegar (UOV) scheme still remains unbroken. Further, this simple and elegant signature scheme boasts small signatures and fast signing times. Arguably, the only drawback to UOV is its rather large public key size. The work of Petzoldt mitigates this by generating the pair $((\mathscr{F},\mathscr{T}),\mathscr{P})$ from a portion of the public key's Macaulay matrix and the map $\mathscr{T}$. By choosing this portion to be easy to store, *i.e.* if it is a cyclic matrix or generated from a pseudo-random number generator, the public key's bit size can be much reduced [37].

A large number of modern schemes are modifications to UOV that are designed to increase efficiency. This is in general hard to do as can be seen from the singularity attack by Ding *et al.* on HIMQ-3, which takes a large amount of its core design from UOV [19]. Out of the nine signature schemes that were accepted to round two of the NIST standardization program, two (LUOV and Rainbow) are based on UOV. Rainbow, originally proposed in 2005, reduces its keysize by forming multiple layers of UOV schemes, where oil variables in a higher layer become vinegar variables in the lower layers [16, 18]. LUOV achieved a reduction in key size by forcing all the coefficients of the public key to either be 0 or 1. In this paper, we will show that such modifications used by LUOV allow for algebraic manipulations that result in an under-determined quadratic system over a much smaller finite field. We will further show that Rainbow and other UOV schemes are immune to such attacks.

1.4 Lifted Unbalanced Oil Vinegar Scheme(LUOV)

The LUOV scheme, as clear from its name, is a modification of the original UOV scheme. Its design was first proposed by Beullens *et al.* in [4]. The core design of LUOV is as follows:

Let $\mathbb{F}_{2^r}$ be a degree r extension of $\mathbb{F}_2$. Let o and v be two positive integers such that $o < v$ and $n = o + v$. The central map $\mathscr{F} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^o$ is a quadratic map whose components $f_1, \ldots, f_o$ are in the form:

$$f_k(\mathbf{x}) = \sum_{i=1}^{v}\sum_{j=i}^{n} \alpha_{i,j,k} x_i x_j + \sum_{i=1}^{n} \beta_{i,k} x_i + \gamma_k,$$

where the coefficients $\alpha_{i,j,k}{}'s$, $\beta_{i,k}{}'s$ and $\gamma_k{}'s$ are chosen randomly from the base field $\mathbb{F}_2$. As in standard UOV, To hide the Oil and Vinegar structure of these polynomials an invertible linear map $\mathscr{T} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^n$ is used to mix the variables. In particular,

the authors of LUOV choose $\mathscr{T}$ in the form:

$$\begin{bmatrix} \mathbf{1}_v & \mathbf{T} \\ \mathbf{0} & \mathbf{1}_o \end{bmatrix}$$

where $\mathbf{T}$ is a $v \times o$ matrix whose entries are from the field $\mathbb{F}_2$. The public key is $\mathscr{P} = \mathscr{F} \circ \mathscr{T}$, where $\mathscr{T}$ and $\mathscr{F}$ are the private keys.

This choice of $\mathscr{T}$, first proposed by Czypek[10], speeds up the key generation and signing process as well as decreases storage requirements. This specific choice of $\mathscr{T}$ does not affect the security of the scheme in comparison to standard UOV due to the fact that for any UOV private key $(\mathscr{F}, \mathscr{T})$ key, there exists a with high probability an equivalent key $(\mathscr{F}', \mathscr{T}')$ such that $\mathscr{T}'$ is in the form chosen by above [42].

The third major modification is the use of the Petzoldt's aforementioned technique to use a pseudo-random number generator to generate both the private key and the public key. This modified key generation algorithm still produces the same distribution of key pairs, and thus the security of the scheme remains unaffected by this modification (assuming that the output of the PRNG is indistinguishable from true randomness). The keys, both public and private, are never directly stored. Each time are wishes to either generate or verify a signature, they are generated from the PRNG.

For the purpose of this paper, much of the details of LUOV are not important. In fact, we will ignore essentially most of the specified structure and focus purely on the "lifted" aspect of the design.

1.5 Our Contributions

We will present a new attack method called the Subfield Differential Attack (SDA). This attack does not rely on the Oil and Vinegar structure of LUOV but merely that the coefficients of the quadratic terms are contained in a small subfield. We will show that the attack will make it impossible for LUOV, as originally presented in the second round of the NIST competition, to fulfill NIST's security level requirements.

For public key $\mathscr{P} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^o$, we assert that with extremely high probability that for a randomly chosen $\mathbf{x}' \in \mathbb{F}_{2^r}^n$ and $\mathbf{y} \in \mathbb{F}_{2^r}^o$ there exists $\bar{\mathbf{x}} \in \mathbb{F}_{2^d}^n$ such that $\mathscr{P}(\mathbf{x}' + \bar{\mathbf{x}}) = \mathbf{y}$, where $\mathbb{F}_{2^d}$ is a subfield of $\mathbb{F}_{2^r}$. By the fact that the coefficients of $\mathscr{P}$ are either 0 or 1 and by viewing $\overline{\mathscr{P}}(\bar{\mathbf{x}}) = \mathscr{P}(\mathbf{x}' + \bar{\mathbf{x}})$ as a system of equations over the smaller field $\mathbb{F}_{2^d}$, we will reduce the forging a signature to solving an under-determined quadratic system over $\mathbb{F}_{2^d}$. The complexity required for such is well under our target. For each proposed set of parameters, we will explicitly apply our attack. We will provide a small toy example. We will explain how UOV and Rainbow are unaffected by our attack. Finally, we will discuss the new parameter sets that LUOV uses in response to SDA.

2 The Subfield Differential Attack on LUOV

2.1 Transforming a LUOV Public by a Differential

The key idea of the attack is to transform the public key, $\mathscr{P}$, into a map over a subfield which is more efficient to work over but still contains a signature for a given message. Namely, maps of the form $\overline{\mathscr{P}} : \mathbb{F}_{2^d}^n \to \mathbb{F}_{2^r}^o$ defined by

$$\overline{\mathscr{P}}(\bar{\mathbf{x}}) = \mathscr{P}(\mathbf{x}' + \bar{\mathbf{x}})$$

where $\mathbf{x}'$ is a random point $\mathbb{F}_{2^r}^n$. We note that for any irreducible polynomial $g(t)$ of degree $r/d = s$,

$$\mathbb{F}_{2^d}[t]/(g(t)) \cong \mathbb{F}_{2^r}.$$

Henceforth, we will represent $\mathbb{F}_{2^r}$ by this quotient ring. Here, $\mathbb{F}_{2^d}$ is embedded as the set of constant polynomials. For more details see [29].

Consider a LUOV public key $\mathscr{P} = \mathscr{F} \circ \mathscr{T} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^o$. Then following the construction of all Oil Vinegar Schemes, $\mathscr{P}$ appears to be a random quadratic system except that all the coefficients are either 0 or 1.

$$\mathscr{P}(\mathbf{x}) = \begin{cases} \tilde{f}_1(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,1} x_i x_j + \sum_{i=1}^{n} \beta_{i,1} x_i + \gamma_1 \\ \tilde{f}_2(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,2} x_i x_j + \sum_{i=1}^{n} \beta_{i,2} x_i + \gamma_2 \\ \qquad\qquad \vdots \\ \tilde{f}_o(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,o} x_i x_j + \sum_{i=1}^{n} \beta_{i,o} x_i + \gamma_o. \end{cases}$$

Randomly chose $\mathbf{x}' \in \mathbb{F}_{2^r}^n$ and define $\overline{\mathscr{P}}(\bar{\mathbf{x}}) = \mathscr{P}(\mathbf{x}' + \bar{\mathbf{x}})$. We see that the k^{th} component of $\overline{\mathscr{P}}$ is of the form:

$$\tilde{f}_k(\mathbf{x}' + \bar{\mathbf{x}}) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k}(x_i' + \bar{x}_i)(x_j' + \bar{x}_j) + \sum_{i=1}^{n} \beta_{i,k}(x_i' + \bar{x}_i) + \gamma_k.$$

Expanding the above and separating the quadratic terms leads to

$$\begin{aligned} \tilde{f}_k(\mathbf{x}' + \bar{\mathbf{x}}) = &\sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k}(x_i' x_j' + x_i' \bar{x}_i + x_j' \bar{x}_j) + \sum_{i=1}^{n} \beta_{i,k}(x_i' + \bar{x}_i) + \gamma_k \\ &+ \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k} \bar{x}_i \bar{x}_j. \end{aligned}$$

On one hand, the coefficients of the quadratic terms in the variables $\bar{\mathbf{x}} = (\bar{x_1}, \ldots, \bar{x_n})$ are still contained in $\mathbb{F}_2$. On the other hand, the x_i' are arbitrary elements of $\mathbb{F}_{2^r}$, and so the linear terms will have coefficients containing all the powers of t. We can thus regroup the above equation in terms of the powers of t, where the quadratic part is confined in the constant term. Meaning, for some linear polynomials $L_{i,k}(\bar{x}_1, \ldots, \bar{x}_n) \in \mathbb{F}_{2^d}[\bar{x}_1, \ldots, \bar{x}_n]$, and quadratic polynomials $Q_k(\bar{x}_1, \ldots, \bar{x}_n) \in \mathbb{F}_{2^d}[\bar{x}_1, \ldots, \bar{x}_n]$, we have that

$$\tilde{f}_k(\mathbf{x}' + \bar{\mathbf{x}}) = \sum_{i=1}^{s-1} L_{i,k}(\bar{x}_1, \ldots, \bar{x}_n) t^i + Q_k(\bar{x}_1, \ldots, \bar{x}_n).$$

2.2 Forging a Signature

Now suppose we want to forge a signature for a message $\mathbf{y} \in \mathbb{F}_{2^r}^o$ where $\mathbf{y} = (y_1, \ldots, y_m)$. Here $y_k = \sum_{i=0}^{s-1} w_{i,k} t^i$ where each $w_{i,k} \in \mathbb{F}_{2^d}$. We will achieve this by solving the system of equations

$$\overline{\mathscr{P}}(\bar{\mathbf{x}}) = \mathbf{y}.$$

This is solving the set of $(s-1)o$ linear equations

$$A = \left\{ L_{i,k}(\bar{x}_1, \ldots, \bar{x}_n) = w_{i,k} : 1 \leq i \leq s-1, 1 \leq k \leq o \right\}$$

and the set of o quadratic equations

$$B = \left\{ Q_k(\bar{x}_1, \ldots, \bar{x}_n) = w_{0,k} : 1 \leq k \leq o \right\}.$$

As A is a random system of linear equations, it has high probability to have rank $(s-1)o$ (or dimension n if $(s-1)o \geq n$). Let S be the solutions space to A. By the Rank Nullity Theorem, the dimension of S is $n-(s-1)o$. We see that our problem thus reduces to solving B over S. That is o quadratic equations in $n-(s-1)o$ variables over the subfield $\mathbb{F}_{2^d}$. Once we find a solution for $\bar{\mathbf{x}}$, the signature is then $\mathbf{x}' + \bar{\mathbf{x}}$ as

$$\mathscr{P}(\mathbf{x}' + \bar{\mathbf{x}}) = \overline{\mathscr{P}}(\bar{\mathbf{x}}) = \mathbf{y}.$$

2.3 The Choice of the Intermediate Field

Now that we know the method of the attack, we need to find the intermediate fields that ensures that $\overline{\mathscr{P}}(\bar{\mathbf{x}}) = \mathbf{y}$ has at least one solution. We wish to compute the probability that, when we define the map $\overline{\mathscr{P}} : \mathbb{F}_{2^d}^n \to \mathbb{F}_{2^r}^o$ as in the prior section, that $\overline{\mathscr{P}}^{-1}(\mathbf{y})$ is nonempty. We will achieve this by heuristically arguing that the quadratic map $\overline{\mathscr{P}}$ acts as a random map. So, we derive the following short lemma:

Lemma 1. *Let A and B be two finite sets and $\mathscr{Q} : A \to B$ be a random map. For each $b \in B$, the probability that $\mathscr{Q}^{-1}(b)$ is non-empty is approximately $1 - e^{-|A|/|B|}$.*

Proof. As the output of each element of A is independent, it is elementary that the probability for there to be at least one $a \in A$ such that $\mathscr{Q}(a) = b$ is

$$1 - \Pr(\mathscr{Q}(\alpha) \neq b, \forall \alpha \in A) = 1 - \prod_{\alpha \in A} \Pr(\mathscr{Q}(\alpha) \neq b) = 1 - \left(1 - \frac{1}{|B|}\right)^{|A|} = 1 - \left(1 - \frac{1}{|B|}\right)^{|B|\frac{|A|}{|B|}}.$$

Using $\lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^n = e^{-1}$, we achieve the desired result.

As a result of this lemma, the probability that $\overline{\mathscr{P}}^{-1}(\mathbf{y})$ is non-empty is approximately $1 - e^{-2^{(dn)-(ro)}}$.

By far the largest cost in the attack is solving the final quadratic system over $\mathbb{F}_{2^d}$. The smaller the d is, the more efficient the cost is. So, we will minimize our choice of d such that the probability of finding a signature is high given our above estimate.

In Tables 6 and 3, we calculate the probability of success on the first guess for $\mathbf{x}'$ for the parameters as originally given for round 2 LUOV (the authors have since changed their parameters due to SDA) [3]. In the astronomically unlikely event that there is no signature, a different guess for $\mathbf{x}'$ can be used. Table 6 is given on parameters designed to reduce the size of signatures. These parameters are used in situations where many signatures are needed. Table 3 is given on parameters designed to reduce the cost of both signatures and public keys. These parameters are used when communicating both signatures and public keys is needed.

NIST Security Level	r	o	v	n	d	Probability of Success
II	8	58	237	295	2	$1-\exp(-2^{126})$
IV	8	82	323	405	2	$1-\exp(-2^{154})$
V	8	107	371	478	2	$1-\exp(-2^{100})$

Table 2. Estimated Probabilities of Success for Parameters Designed to Minimize the Size of the Signature

NIST Security Level	r	o	v	n	d	Probability of Success
II	48	43	222	265	8	$1-\exp(-2^{56})$
IV	64	61	302	363	16	$1-\exp(-2^{1904})$
V	80	76	363	439	16	$1-\exp(-2^{944})$

Table 3. Estimated Probabilities of Success for Parameters Designed to Minimize the Size of the Signature and Public Key

3 Complexity of the Attack

While there is some slight overhead cost in computing $\overline{\mathscr{P}}(\overline{\mathbf{x}})$ and solving the linear system, the vast majority of the complexity is solving the quadratic system of $n-(s-1)o$ variables and o equations over $\mathbb{F}_{2^d}$. Hence, to evaluate the effectiveness of our attack we will compute the complexity of finding a single solution to this quadratic system, which we will measure with the number of field multiplications. As this is a underdetermined system, the most effective strategy is first to use the method of Thomae and Wolf [41] to transform it by a linear change of variables to a determined system with fewer equations than before.

3.1 Statement and Results of Thomae and Wolf

Theorem 1 (Thomae and Wolf). *By a linear change of variables, the complexity of solving an underdetermined quadratic system of m equations and $n=\omega m$ variables can be reduced to solving a determined quadratic system of $m-\lfloor\omega\rfloor+1$ equations.*

Further, if $\lfloor\omega\rfloor | m$ then the complexity can be further reduced to solving a determined quadratic system of $m - \lfloor\omega\rfloor$ equations.

We calculate what these new determined systems will be in Table 4 for the various parameter sets representing each system as (number of variables) × (number of equations). The complexity will depend on direct methods of solving these systems of equations.

Table and Security	Finite Field	Original System	New System
(2, II)	$\mathbb{F}_{2^2}$	58×121	56×56
(2, IV)	$\mathbb{F}_{2^2}$	82×159	81×81
(2, V)	$\mathbb{F}_{2^2}$	107×157	106×106
(3, II)	$\mathbb{F}_{2^8}$	43×50	42×42
(3, IV)	$\mathbb{F}_{2^{16}}$	61×180	60×60
(3, V)	$\mathbb{F}_{2^{16}}$	76×135	75×75

Table 4. Determined Systems to Solve after Thomae and Wolf

3.2 Solving the Determined Systems

To find a solution to one of these determined systems, the best method is to use what is called the hybrid approach [1, 2] which involves repeatedly fixing some of the values of the variables and then performing a direct attack on the new overdetermined system until a solution is found. The amount of variables guessed for depends on the algorithm and the finite field involved with a smaller finite field leading to more variables being guessed for.

The two main contenders for the best algorithm to use are one of the family of XL (eXtended Linearization) algorithms proposed by Courtois *et al.* [9] and either the F4/F5 algorithms proposed by Faugère [22, 23] or algorithms developed from these two. In our case both will give comparable results though we will follow the work of Yet *et al.* [45] and favor the former using Wiedemann XL, the reason why we will explain shortly.

Let us give a brief description of the XL algorithm which, for simplicity, we will give for the case of quadratic systems. Let $\mathscr{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ by a given quadratic system we want to solve where $\mathscr{P} = (p_1, \ldots, p_m)$. As in our case we will be working with overdetermined systems, we can assume that there will be at most one solution as can be justified by Lemma 1. We will denote a monomial $x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ by $\mathbf{x}^{\mathbf{b}}$ where $\mathbf{b} = (b_1, \ldots, b_n)$ and $|\mathbf{b}| = b_1 + b_2 + \cdots + b_n$. For a given natural number D, let us denote by $T^{(D)} = \{\mathbf{x}^{\mathbf{b}} : |\mathbf{b}| \leq D\}$ the set of all degree D or lower monomials. We note that $|T^{(D)}| = \binom{n+D}{D}$ as was shown in [9] but as we only seek a solution in the field $\mathbb{F}_q$ we can reduce this by equating $x_i^q = x_i$ leading to

$$|T^{(D)}| = [t^D] \frac{(1-t^q)^n}{(1-t)^{n+1}}$$

where $[t^D]g(t)$ is the coefficient of t^D in the series expansion $g(t)$ [44].

One begins by extending $\mathscr{P}$ to the set of relations $R^{(D)} = \{\mathbf{x}^{\mathbf{b}} p_i(\mathbf{x}) = 0 : 1 \leq i \leq m, \mathbf{x}^{\mathbf{b}} \in T^{(D-2)}\}$. Let us denote by $M^{(D)}$ the Macaulay matrix for $R^{(D)}$. One performs linear algebra techniques to attempt to solve $M^{(D)}$, and provided D is large enough one will either find a solution, a univariate polynomial for one of the variables which then can be solved for, or a contradiction. Obviously, the smallest such D will allow the lowest complexity in working with $M^{(D)}$ as the size of $M^{(D)}$ depends on D. We will denote this by D_0 which is called the operating degree of XL. Yeh *et al.* [45] stated that for random quadratic systems (which UOV systems behave like) over small fields (when the operating degree is larger than the size of the field) we will have

$$D_0 = \min\left\{d : [t^d]\frac{(1-t^q)^n(1-t^2)^m}{(1-t)^{n+1}(1-t^{2q})^m} \leq 0\right\}.$$

For larger fields we will instead have

$$D_0 = \min\left\{d : [t^d](1-t)^{m-n-1}(1+t)^m < 0\right\}.$$

The Macaulay matrix $M^{(D_0)}$ is a sparse matrix with total weight approximately equal to $|R^{(D_0)}|n^2/2$. This is one of the advantages of using XL as it allows one to solve the linear system by using the (block) Wiedemann matrix solver [8] in approximately $\frac{3}{2}|R^{(D_0)}||T^{(D_0)}|n^2$ field multiplications. By randomly discarding rows (most of of which are nonessential for solving the system) until there are $|T^{(D_0)}|$ left the number of field multiplications becomes $\frac{3}{2}|T^{(D_0)}|^2 n^2$ [43]. As $|T^{(D_0)}| \leq \binom{n+D_0}{D_0}$ and $\frac{n^2}{2} \approx \binom{n}{2}$ we can estimate this as $3 \times \binom{n+D_0}{D_0} \times \binom{n}{2}$.

Returning our focus to the determined systems we are dealing with in attacking LUOV, if we denote the number of variables we are guessing for as k and $D_0^{(k)}$ as the calculated operating degree after guessing for those variables, we have the following theorem with the additional factor of q^k accounting for the necessary repeated attempts due to the potential of incorrect guessing.

Theorem 2. *The complexity in terms of field multiplications of performing the XL algorithm on a determined quadratic system of m equations over a finite field of size q is*

$$Complexity_{XL} = min_k\left\{q^k \times 3 \times \binom{m-k+D_0^{(k)}}{D_0^{(k)}}^2 \times \binom{m-k}{2}\right\}.$$

While there are other, more sophisticated, versions of XL like mutant XL and its sub-variants [11, 31, 32, 33] that can also perform well in certain situations, the Wiedemann algorithm offers parallel compatibility and cheaper memory cost [7] that, along with its computation time, makes Wiedemann XL better than the other variants of XL for our case. It is for this same reason that we prefer Wiedemann XL over F4 and F5.

Now let us briefly describe our reasoning for preferring Wiedemann XL to either F4 or F5 in estimating the complexity of the attack. F4 [22] is an improvement of Buchberger's algorithm [6] for generating a Groebner for the ideal generated by the

quadratic system $\mathscr{P}$. F4 also works with linear algebra techniques with a Macaulay matrix which allows it to do reduction steps in parallel to compute normal forms, eventually generating a Groebner basis. Thus its complexity will be determined the largest size of the matrix involved and the linear algebra cost in working with that matrix.

The size of the matrix will be determined by what is called the degree of regularity which is the degree at which the first non-trivial relation from the original polynomials $p_1, \cdots, p_m$ occurs. The trivial relations are $p_i^h p_j^h - p_j^h p_i^h = 0$ and $p_i^q - p_i = 0$. All others are nontrivial. We will denote this by D_{reg}. As F4 will have to deal with polynomial of degree D_{reg} [14], the size of the matrix will be roughly $|T^{(D_{reg})}| = \binom{n+D_{reg}}{D_{reg}}$ rows and columns.

The F5 algorithm [23] is an improvement on the F4 algorithm in that it too uses linear algebra techniques to construct a Groebner for the quadratic system. With the use of what Faugère calls signatures of the polynomials one can perform fewer reduction steps than F4. This is because some of the row reductions in F4 represent reductions to 0 meaning they are essentially useless in constructing the Groebner basis. The F5 algorithm uses the signatures to know beforehand not do do these reductions. We note that we cannot find independent implementation of the F5 algorithm which meets the originally claimed levels of efficiency, and that the original "proof" of the termination of F5 was flawed. It was not until 2012 when Galkin [24] proved the termination (in fact in a more general case than originally proposed). There has been much research conducted on F5 inspired signature-based Groebner algorithms [21]. However, these improvements (some of which were not based on complexity at all but the issue of termination) are not large enough to overcome the largest determining factor in their complexity: the size of the Macaulay matrix involved. As the degree of regularity for F5 and F4 are the same D_{reg} [45], the matrix that F5 and F5 inspired algorithms will be working with is essentially the same size as F4 but having fewer rows due to the use of signatures.

The complexity for F4/F5 will then be approximately $\binom{n+D_{reg}}{D_{reg}}^{\omega}$ where $2 \leq \omega \leq 3$ is the complexity exponent of matrix multiplication. ω is likely to be about $\log_2(7) \approx 2.8$ though may be as low as 2.3727 [45]. We note that there has been work on improving the linear algebra cost involved in Groebner basis calculations due to the special shape of the matrices involved such as the GBLA library [5]. However, due to the fact that the matrix involved in XL is more sparse than that in F4 or F5 [45], the linear algebra for F4/F5 is more costly than that for Wiedemann XL and the memory size is greater for F4/F5 as well provided the size of the matrices are relatively close which happens when D_0 is very close to D_{reg} [45]. It is known that $D_{reg} \leq D_0$ so there will be fewer rows needed to work with when using F4 or F5. Further, for certain polynomial systems with specific (even if hidden) structure like HFE and its variants, D_{reg} may be much smaller for which there is much research [13, 14, 17, 20, 26]. In these cases, an F4/F5 type algorithm is the best to use. However, Yeh *et al.* [45] has shown that for random overdetermined systems, like the ones we are attacking after we fix some variables, $D_0 - D_{reg}$ is most often ≤ 1 and in many cases 0. They give D_{reg} for quadratic systems over small fields as

$$D_{reg} = \min\left\{d : [t^d]\frac{(1-t^q)^n(1-t^2)^m}{(1-t)^n(1-t^{2q})^m} < 0\right\}$$

and for larger fields

$$D_{reg} = \min\left\{d : [t^d](1-t)^{m-n}(1+t)^m < 0\right\}.$$

As an example Figure 1 shows both D_0 and D_{reg} after the different choices of variables to fix for the system of 56 variables and 56 equations over $\mathbb{F}_{2^2}$. We see that the difference is never more than 1 and often is 0. Thus we will use Theorem 2's estimate for the complexity of the attack using Wiedemann XL.

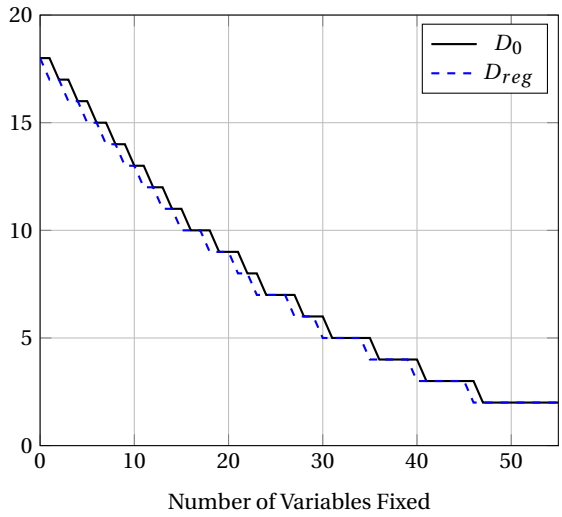

Fig. 1. D_0 and D_{reg} for the system with 56-k Variables and 56 Equations over $\mathbb{F}_{2^2}$

3.3 Calculating the Complexity

As an example, let's estimate the complexity of forging a signature for a LUOV public key with parameters $r = 8, o = 58, v = 237$ using Wiedemann XL. We need only to focus on solving the quadratic over the intermediate field as additional overhead is very small. As mentioned before, the optimal choice for the intermediate field is $\mathbb{F}_{2^2}$. The resulting quadratic system over this smaller field has $o = 58$ equations and $n - (s-1)o = 121$ variables. As $\lfloor 121/58 \rfloor = 2$ which divides 58, we can use the stronger version of Theorem 1. So, the complexity is reduced to solving a determined system of $58 - 2 = 56$ equations.

We search through the complexities of the XL algorithm for the various choices of k, and we find the smallest is when $k = 31$. In this case,

$$\frac{(1-t^q)^{m-k}(1-t^2)^m}{(1-t)^{m-k+1}(1-t^{2q})^m}, = 1 + 26t + 295t^2 + 1820t^3 + 5610t^4 - 1560t^5 + \cdots.$$

So the first power of t with a non-positive coefficient is t^5. Thus, $D_0^{(31)} = 5$.

Finally, we compute the complexity as

$$4^{31} \times 3 \times \binom{56-31+5}{5}^2 \times \binom{56-31}{2} = 842885418247230178100716240896000 \approx 2^{107}.$$

In Table 5 we compute the complexity for the various parameters found in the original round 2 submission. We round up the given log base 2 complexity.

Table and Security	Finite Field	Original System	New System	# of Guesses	$D_0^{(k)}$	Log_2 Complexity
(2, II)	$\mathbb{F}_{2^2}$	58×121	56×56	31	5	107
(2, IV)	$\mathbb{F}_{2^2}$	82×159	81×81	38	8	146
(2, V)	$\mathbb{F}_{2^2}$	107×157	106×106	51	9	184
(3, II)	$\mathbb{F}_{2^8}$	43×50	42×42	3	19	135
(3, IV)	$\mathbb{F}_{2^{16}}$	61×180	60×60	2	31	202
(3, V)	$\mathbb{F}_{2^{16}}$	76×135	75×75	2	38	244

Table 5. Complexity in Terms of Number of Field Multiplications

Recalling that NIST requires complexity $(2^{146}, 2^{210}, 2^{272})$ for security levels (II, IV, V) respectively, we see that LUOV fails to meet the security level requirements in all parameter sets given for their targeted security.
The two schemes which claim to be of Level II security do not even satisfy the Level I security, which is supposed to be 2^{143}.

3.4 Toy Example

Let $o = 2$, $v = 8$, and $n = 10$. The size of the large extension field chosen by the public key generator will be $2^8 = 256$. In the attack, we will use our small field $\mathbb{F}_{2^2}$ denoting its elements by $\{0, 1, w_1, w_2\}$. We will then represent the field $\mathbb{F}_{2^8}$ by $\mathbb{F}_{2^2}[t]/f(t)$ where $f(t) = t^4 + t^2 + w_1 t + 1$.

Consider the LUOV public key $\mathscr{P} : \mathbb{F}_{2^8}^n \to \mathbb{F}_{2^8}^o$, where for simplicity sake, it will be homogeneous of degree two:

$$\begin{aligned}
\tilde{f}_1(\mathbf{x}) =& x_1x_4 + x_1x_5 + x_1x_6 + x_1x_7 + x_1x_8 + x_1x_9 + x_2x_4 + x_2x_6 + x_2x_9 + x_3^2 \\
&+ x_3x_6 + x_3x_7 + x_3x_{10} + x_4^2 + x_4x_7 + x_4x_8 + x_4x_9 + x_4x_{10} + x_5x_6 + x_6x_{10} \\
&+ x_7^2 + x_7x_8 + x_7x_9 + x_8x_9 + x_8x_{10} + x_9^2 + x_9x_{10} \\
\tilde{f}_2(\mathbf{x}) =& x_1x_3 + x_1x_4 + x_1x_5 + x_1x_9 + x_2x_3 + x_2x_6 + x_2x_7 + x_2x_9 + x_3^2 + x_3x_4 \\
&+ x_3x_5 + x_3x_6 + x_3x_7 + x_3x_9 + x_4^2 + x_4x_5 + x_4x_6 + x_4x_7 + x_4x_{10} + x_5^2 \\
&+ x_5x_6 + x_5x_7 + x_5x_8 + x_5x_{10} + x_6x_7 + x_7x_9 + x_9x_{10} + x_{10}^2
\end{aligned}$$

We will attempt to find a signature for the message:

$$\mathbf{y} = \begin{bmatrix} w_1 t^3 + w_2 t^2 + w_2 t \\ w_2 t^3 + w_2 t^2 + t \end{bmatrix}$$

First, we randomly select our $\mathbf{x}'$ as

$$\mathbf{x}' = \begin{bmatrix} t^3 + w_2 t \\ w_1 t^3 + w_2 t^2 + w_2 t \\ t^3 + t + 1 \\ w_2 t^2 + w_1 \\ t^3 + t^2 + 1 \\ w_2 t^3 + t^2 + w_2 t + w_2 \\ w_1 t^3 + w_2 t + w \\ w_1 t^2 + w_2 t + 1 \\ t^3 + w_2 t + w_1 \\ w_2 t + w_2 \end{bmatrix}$$

We then calculate $\mathscr{P}(\mathbf{x}' + \bar{\mathbf{x}})$ and represent it as a polynomial of t:

$$\begin{aligned}
\tilde{f}_1(\mathbf{x}' + \bar{\mathbf{x}}) = &(\bar{x}_1 + w_1\bar{x}_2 + \bar{x}_3 + w_1\bar{x}_5 + w_2\bar{x}_6 + \bar{x}_7 + w_1\bar{x}_8 + \bar{x}_9 + w_2\bar{x}_{10})\,t^3 \\
&+ (\bar{x}_1 + w_1\bar{x}_2 + \bar{x}_3 + \bar{x}_4 + \bar{x}_5 + w_1\bar{x}_6 + \bar{x}_7 + w_2\bar{x}_8 + w_1\bar{x}_9)\,t^2 \\
&+ (w_2\bar{x}_3 + w_1\bar{x}_6 + w_1\bar{x}_7 + w_2\bar{x}_9 + w_1\bar{x}_{10})\,t \\
&+ Q_1(\bar{x}_1, \ldots, \bar{x}_n) \\
\tilde{f}_2(\mathbf{x}' + \bar{\mathbf{x}}) = &(\bar{x}_1 + \bar{x}_2 + w_1\bar{x}_3 + \bar{x}_5 + \bar{x}_8)\,t^3 \\
&+ (w_1\bar{x}_1 + \bar{x}_2 + \bar{x}_6 + \bar{x}_8 + w_2\bar{x}_9 + w_1\bar{x}_{10})\,t^2 \\
&+ (w_1\bar{x}_1 + w_1\bar{x}_2 + w_2\bar{x}_3 + \bar{x}_4 + w_1\bar{x}_5 + \bar{x}_6 + w_1\bar{x}_7 + \bar{x}_9 + w_2\bar{x}_{10})\,t \\
&+ Q_2(\bar{x}_1, \ldots, \bar{x}_n),
\end{aligned}$$

where $Q_1(\bar{x}_1, \ldots, \bar{x}_n)$ and $Q_2(\bar{x}_1, \ldots, \bar{x}_n)$ are quadratic polynomials from $\mathbb{F}_{2^2}[\bar{x}_1, \ldots, \bar{x}_n]$. By comparing the coefficients of t^3, t^2, t^1 and assuming $\mathscr{P}(\mathbf{x}' + \bar{\mathbf{x}}) = \mathbf{y}$, we arrive at a system of linear equations over $\mathbb{F}_{2^2}$. This can be represented by a matrix equation $\mathbf{A}\mathbf{x} = \mathbf{y}$. In our case, this is the following:

$$\begin{bmatrix}
1 & w_1 & 1 & 0 & w_1 & w_2 & 1 & w_1 & 1 & w_2 \\
1 & w_1 & 1 & 1 & 1 & w_1 & 1 & w_2 & w_1 & 0 \\
0 & 0 & w_2 & 0 & 0 & w_1 & w_1 & 0 & w_2 & w_1 \\
1 & 1 & w_1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
w_1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & w_2 & w_1 \\
w_1 & w_1 & w_2 & 1 & w_1 & 1 & w_1 & 0 & 1 & w_2
\end{bmatrix}
\begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \\ \bar{x}_5 \\ \bar{x}_6 \\ \bar{x}_7 \\ \bar{x}_8 \\ \bar{x}_9 \\ \bar{x}_{10} \end{bmatrix}
= \begin{bmatrix} w_1 \\ w_2 \\ w_2 \\ w_2 \\ w_2 \\ 1 \end{bmatrix}$$

The solution space for the equation above has dimension 4 over $\mathbb{F}_{2^2}$, as we would expect it to be $n-(s-1)o=4$. Thus, there are only $(2^2)^4=2^8$ possible choices for $\bar{\mathbf{x}}$. A quick search through these finds the signature

$$\sigma = \begin{bmatrix} t^3+w_2t+1 \\ w_1t^3+w_2t^2+w_2t+w_1 \\ t^3+t+w_2 \\ w_2t^2 \\ t^3+t^2+1 \\ w_2t^3+t^2+w_2t+1 \\ w_1t^3+w_2t+w_1 \\ w_1t^2+w_2t+1 \\ t^3+w_2t+1 \\ w_2t \end{bmatrix}$$

4 The Inapplicability of the Subfield Differential Attack on Unbalanced Oil Vinegar

Now, let us discuss why the Subfield Differential Attack does not work on Unbalanced Oil Vinegar or Rainbow. Let $\mathscr{P}:\mathbb{F}_{q^r}^n \to \mathbb{F}_{q^r}^o$ be either a UOV public key or a Rainbow public key. Let us assume that $\mathbb{F}_{q^r}$ contains a non-trivial subfield $\mathbb{F}_{q^d}$. Again, construct the differential $\mathbf{x}'+\bar{\mathbf{x}}$ with $\mathbf{x}'\in\mathbb{F}_{q^r}$ and $\bar{\mathbf{x}}\in\mathbb{F}_{q^d}$, and evaluate the public key at the differential $\overline{\mathscr{P}}(\bar{\mathbf{x}})=\mathscr{P}(\mathbf{x}'+\bar{\mathbf{x}})$. In the k^{th} component of $\overline{\mathscr{P}}$, we have that

$$\bar{f}_k(\mathbf{x}'+\bar{\mathbf{x}}) = \sum_{i=1}^{n}\sum_{j=i}^{n}\alpha_{i,j,k}(x_i'+\bar{x}_i)(x_j'+\bar{x}_j)+\sum_{i=1}^{n}\beta_{i,k}(x_i'++\bar{x}_i)+\gamma_k.$$

Note that there are no restrictions on the coefficients, $\alpha_{i,j,k},\beta_{i,k}$ and γ_k as they are randomly chosen from $\mathbb{F}_{q^r}$. If we multiply the polynomial out, then we get

$$\begin{aligned}\tilde{f}_k(\mathbf{x}'+\mathbf{x}) = &\sum_{i=1}^{n}\sum_{j=i}^{n}\alpha_{i,j,k}(x_i'x_j'+x_i'\bar{x}_i+x_j'\bar{x}_j)+\sum_{i=1}^{n}\beta_{i,k}(x_i'+\bar{x}_i)+\gamma_k \\ &+\sum_{i=1}^{n}\sum_{j=i}^{n}\alpha_{i,j,k}\bar{x}_i\bar{x}_j.\end{aligned}$$

The quadratic terms' coefficients will not be contained in the subfield $\mathbb{F}_{q^d}$. Thus, instead of having a clear separation of $(s-1)o$ linear polynomials and o quadratic polynomials over $\mathbb{F}_{2^d}$ as before for a LUOV public key, we instead have $s*o$ quadratic polynomials over $\mathbb{F}_{q^d}$. Thus it is not more efficient to direct attack than simply having o quadratic polynomials over $\mathbb{F}_{q^r}$, and so viewing the field as a quotient ring does not help for UOV or Rainbow. So the SDA attack does not apply to these schemes.

5 New Parameter Sets for LUOV in Response to SDA

We note that in response to the SDA attack, the authors of LUOV have submitted new parameters sets designed to avoid the existence of a sufficiently large intermediate field to perform SDA. In particular, they chose the extension $\mathbb{F}_{2^r}$ where r is a

prime number. This means that only the trivial subfield $\mathbb{F}_2$ exists which is not large enough to find a signature over given their new parameters. Table 5 lists these new parameters.

Name	NIST Security Level	r	o	v	n
LUOV-7-57-197	I	7	57	197	254
LUOV-7-83-283	III	7	83	283	366
LUOV-7-110-374	V	7	110	374	484
LUOV-47-42-182	I	47	42	182	224
LUOV-61-60-261	III	61	60	261	321
LUOV-79-76-341	V	79	76	341	417

Table 6. The New Parameter Sets for LUOV

These new modifications are very new and untested. There is a good possibility that a more robust SDA variant utilizing special subsets of $\mathbb{F}_{2^r}$ instead of just subfields could handle a wider variety of parameters, including the current parameters of LUOV. Further research is needed in this area.

6 Conclusion

We proposed a new attack to a NIST round 2 candidate LUOV. All the parameters originally set for round 2 LUOV were broken according to the NIST standards. SDA only uses the basic structure of field extensions which is the core idea of LUOV. The idea of our attack is simple, however it has great potential. Its simple structure leaves room for improvement and modification to handle more cases more efficiently. Furthermore, one can see that the attack does not depend on the design of the central map. It can be applied to other schemes with a lifted structure and solving lifted quadratic systems in general. We believe that future study of SDA is warranted.

7 Acknowledgments

First we would like to thank Bo-Yin Yang for useful discussions, in particular, on the complexity analysis. Second, We would also like to thank partial support of NSF (Grant: #CNS-1814221) and NIST, and J. Ding would like to thank the TAFT Research Center for many years' support. Finally, we are grateful for the comments of the referees helping us improve the quality of this paper.

Bibliography

[1] Luk Bettale, Jean-Charles Faugere, and Ludovic Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2009.

[2] Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Solving polynomial systems over finite fields: improved analysis of the hybrid approach. In *Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, pages 67–74, 2012.

[3] W Beullens, B Preneel, A Szepieniec, and F Vercauteren. Luov signature scheme proposal for nist pqc project (round 2 version), 2018.

[4] Ward Beullens and Bart Preneel. Field lifting for smaller uov public keys. In *International Conference on Cryptology in India*, pages 227–246. Springer, 2017.

[5] Brice Boyer, Christian Eder, Jean-Charles Faugère, Sylvian Lachartre, and Fayssal Martani. Gbla: Gröbner basis linear algebra package. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 135–142, 2016.

[6] Bruno Buchberger. A theoretical basis for the reduction of polynomials to canonical forms. *ACM SIGSAM Bulletin*, 10(3):19–29, 1976.

[7] Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, and Bo-Yin Yang. Solving quadratic equations with xl on parallel architectures - extended version. Cryptology ePrint Archive, Report 2016/412, 2016. https://eprint.iacr.org/2016/412.

[8] Don Coppersmith. Solving homogeneous linear equations over gf(2) via block wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994.

[9] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 392–407. Springer, 2000.

[10] Peter Czypek. *Implementing Multivariate Quadratic Public Key Signature Schemes on Embedded Devices*. PhD thesis, Citeseer, 2012.

[11] Jintai Ding, Johannes Buchmann, Mohamed Saied Emam Mohamed, WSAE Mohamed, and Ralf-Philipp Weinmann. Mutant xl. talk at the first international conference on symbolic computation and cryptography (scc 2008), 2008.

[12] Jintai Ding, Jason E. Gower, and Dieter Schmidt. *Multivariate Public Key Cryptosystems*, volume 25 of *Advances in Information Security*. Springer, 2006.

[13] Jintai Ding and Timothy J Hodges. Inverting hfe systems is quasi-polynomial for all fields. In *Annual Cryptology Conference*, pages 724–742. Springer, 2011.

[14] Jintai Ding and Thorsten Kleinjung. Degree of regularity for hfe-. *IACR Cryptology ePrint Archive*, 2011:570, 2011.

[15] Jintai Ding and Albrecht Petzoldt. Current state of multivariate cryptography. *IEEE Security & Privacy*, 15(4):28–36, 2017.

[16] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In *International Conference on Applied Cryptography and Network Security*, pages 164–175. Springer, 2005.

[17] Jintai Ding and Bo-Yin Yang. Degree of regularity for hfev and hfev. In *International Workshop on Post-Quantum Cryptography*, pages 52–66. Springer, 2013.

[18] Jintai Ding, Bo-Yin Yang, Chia-Hsin Owen Chen, Ming-Shing Chen, and Chen-Mou Cheng. New differential-algebraic attacks and reparametrization of rainbow. In *Applied Cryptography and Network Security, 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings*, pages 242–257, 2008.

[19] Jintai Ding, Zheng Zhang, Joshua Deaton, and Vishakha. The singularity attack to the multivariate signature scheme himq-3. Cryptology ePrint Archive, Report 2019/895, 2019. https://eprint.iacr.org/2019/895.

[20] Vivien Dubois and Nicolas Gama. The degree of regularity of hfe systems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 557–576. Springer, 2010.

[21] Christian Eder and Jean-Charles Faugère. A survey on signature-based algorithms for computing gröbner bases. *Journal of Symbolic Computation*, 80:719–784, 2017.

[22] Jean-Charles Faugere. A new efficient algorithm for computing gröbner bases (f4). *Journal of pure and applied algebra*, 139(1-3):61–88, 1999.

[23] Jean Charles Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero (f 5). In *Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 75–83, 2002.

[24] Vasily Galkin. Termination of original f5, 2012.

[25] Patrick Gallagher. Digital signature standard (dss). *Federal Information Processing Standards Publications, volume FIPS*, pages 186–3, 2013.

[26] Xin Jiang, Jintai Ding, and Lei Hu. Kipnis-shamir attack on hfe revisited. In *International Conference on Information Security and Cryptology*, pages 399–411. Springer, 2007.

[27] David S Johnson and Michael R Garey. *Computers and intractability: A guide to the theory of NP-completeness*. WH Freeman, 1979.

[28] Aviad Kipnis and Adi Shamir. Cryptanalysis of the oil and vinegar signature scheme. In *Annual International Cryptology Conference*, pages 257–266. Springer, 1998.

[29] Rudolf Lidl and Harald Niederreiter. *Finite fields*, volume 20. Cambridge university press, 1997.

[30] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 419–453. Springer, 1988.

[31] Mohamed Saied Emam Mohamed, Daniel Cabarcas, Jintai Ding, Johannes Buchmann, and Stanislav Bulygin. Mxl 3: an efficient algorithm for computing gröbner bases of zero-dimensional ideals. In *International Conference on Information Security and Cryptology*, pages 87–100. Springer, 2009.

[32] Mohamed Saied Emam Mohamed, Jintai Ding, Johannes Buchmann, and Fabian Werner. Algebraic attack on the mqq public key cryptosystem. In *International Conference on Cryptology and Network Security*, pages 392–401. Springer, 2009.

[33] Mohamed Saied Emam Mohamed, Wael Said Abd Elmageed Mohamed, Jintai Ding, and Johannes Buchmann. Mxl2: Solving polynomial equations over gf (2) using an improved mutant strategy. In *International Workshop on Post-Quantum Cryptography*, pages 203–215. Springer, 2008.

[34] National Institute of Standards and Technology. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. Technical report, National Institute of Standards and Technology, 2017.

[35] Jacques Patarin. Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt'88. In *Annual International Cryptology Conference*, pages 248–261. Springer, 1995.

[36] Jacques Patarin. The oil and vinegar algorithm for signatures. In *Dagstuhl Workshop on Cryptography, 1997*, 1997.

[37] Albrecht Petzoldt, Stanislav Bulygin, and Johannes Buchmann. Linear recurring sequences for the uov key generation. In *International Workshop on Public Key Cryptography*, pages 335–350. Springer, 2011.

[38] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[39] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[40] William Stallings. *Cryptography and Network Security, 4/E*. Pearson Education India, 2006.

[41] Enrico Thomae and Christopher Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, pages 156–171. Springer, 2012.

[42] Christopher Wolf and Bart Preneel. Equivalent keys in multivariate quadratic public key systems. *Journal of Mathematical Cryptology*, 4(4):375–415, 2011.

[43] Bo-Yin Yang, Chia-Hsin Owen Chen, Daniel J. Bernstein, and Jiun-Ming Chen. Analysis of QUAD. In *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, pages 290–308. Sringer, 2007.

[44] Bo-Yin Yang and Jiun-Ming Chen. Theoretical analysis of xl over small fields. In *Australasian Conference on Information Security and Privacy*, pages 277–288. Springer, 2004.

[45] Jenny Yuan-Chun Yeh, Chen-Mou Cheng, and Bo-Yin Yang. Operating degrees for xl vs. f4/f5 for generic $\mathscr{MQ}$ with number of equations linear in that of variables. In *Number Theory and Cryptography*, pages 19–33. Springer, 2013.