The Nested Subset Differential Attack

A Practical Direct Attack Against LUOV Which Forges a Signature Within 210 Minutes

Jintai Ding[1], Joshua Deaton[2(✉)], Vishakha[2], and Bo-Yin Yang[3]

[1] Tsinghua University, Beijing, China
[2] University of Cincinnati, Cincinnati, OH, USA
{deatonju,sharmav4}@mail.uc.edu
[3] Academia Sinica, Taipei, Taiwan
byyangat@iis.sinica.edu.tw

Abstract. In 2017, Ward Beullens *et al.* submitted Lifted Unbalanced Oil and Vinegar [3], which is a modification to the Unbalanced Oil and Vinegar Scheme by Patarin. Previously, Ding *et al.* proposed the Subfield Differential Attack [22] which prompted a change of parameters by the authors of LUOV for the second round of the NIST post quantum standardization competition [4].

In this paper we propose a modification to the Subfield Differential Attack called the Nested Subset Differential Attack which fully breaks half of the parameter sets put forward. We also show by experimentation that this attack is practically possible to do in under 210 min for the level I security parameters and not just a theoretical attack. The Nested Subset Differential attack is a large improvement of the Subfield differential attack which can be used in real world circumstances. Moreover, we will only use what is called the "lifted" structure of LUOV, and our attack can be thought as a development of solving "lifted" quadratic systems.

1 Introduction

1.1 Signature Schemes, Post-quantum Cryptography and the NIST Post Quantum Standardization

Signature schemes allow one to digitally sign a document. These were first theoretically proposed by Whitfield Diffie and Martin Hellman using public key cryptography in [12]. The first and still most commonly used scheme is that of RSA made by Rivest, Shamir, and Adleman [35]. As technology and long distance communication become increasingly more a part of everyone's life, it becomes vital that one can verify who sent them a message and sign off on any message they intend to send. However, quantum computers utilizing Shor's algorithm threaten the security of the RSA scheme and many others now in use [37]. With the recent progress of building quantum computers, post-quantum cryptography able to resist quantum attacks has become a central research topic [1,7,8,30]. In 2016, NIST put out a call for proposals for post-quantum cryptosystems for

A. Canteaut and F.-X. Standaert (Eds.): EUROCRYPT 2021, LNCS 12696, pp. 329–347, 2021.
https://doi.org/10.1007/978-3-030-77870-5_12

standardization. These cryptosystems, though using classical computing in their operations, would resist quantum attacks [31]. We are currently in the third round of the "competition," with many different types of schemes being proposed. Multivariate cryptography is one family of post-quantum cryptosystems which is promising to resist quantum attacks [13,16].

1.2 Multivariate Cryptography

Public key encryption and signature schemes rely on a trapdoor function, one which is very difficult to invert except if one has special knowledge about the specific function. Multivariate cryptography bases its trapdoors on the difficulty of solving a random system of m polynomials in n variables over a finite field. For efficiency these polynomials are generally of degree 2. This has been proven to be NP hard [25], and thus is a good candidate for a public key cryptosystem. Moreover, working over these finite fields is often more efficient than older number-theory based methods like RSA. The difficulty lies in the fact that, as these systems must be invertible for the user and thus require a trapdoor, they are not truly random and must have a specific form which undermines the supposed NP hardness of solving them. Generally their special form is hidden by composition by invertible affine maps. Though there are interesting and practical multivariate encryption schemes [17,38,39], multivariate schemes are better known for simple and efficient signature scheme.

The first real breakthrough for multivariate cryptography was the MI or C^* cryptosystem proposed by Matsumoto and Imai in 1988 [29]. Their insight was to use the correspondence ψ between a n dimensional vector space k^n over a finite field k and a n degree extension K over k. They constructed their univariate trapdoor function $\mathscr{F} : K \to K$ over the large field which they were able to solve due to its special shape, and then composed it with two invertible affine maps $\mathscr{S}, \mathscr{T} : k^n \to k^n$ hiding its structure. Their public key is then $\mathscr{P} = \mathscr{S} \circ \psi \circ \mathscr{F} \circ \psi^{-1} \circ \mathscr{T}$. Though broken today, the MI cryptosystem is the inspiration for all "big field" schemes which have their trapdoor over a larger field. But the attack against MI is the inspiration for what are called oil and vinegar schemes, which LUOV is a extension of. The Linearization Equation Attack was developed by Patarin [32]. 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 the above equation 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 $\mathbf{x}$. So a quadratic problem becomes linear and thus easy to solve.

1.3 Oil and Vinegar Schemes

Inspired by the Linearization Equation Attack, Patarin introduced the Oil and Vinegar scheme [33]. The key idea is to reduce the problem of solving a quadratic system of equations into solving a linear system by separating the variables into two types, the vinegar variables which can be guessed for and the oil variables which will be solved for. Let $\mathbb{F}$ be a (generally small) finite field, m and v be two integers, and $n = m + v$. The central map $\mathscr{F} : \mathbb{F}^n \to \mathbb{F}^m$ is a quadratic map whose components $f_1, \ldots, f_m$ are in the form

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

where each coefficient is in $\mathbb{F}$. Here the set of variables $V = \{x_1, \ldots, x_v\}$ are called the vinegar variables, and the set $O = \{x_{v+1}, \ldots, x_n\}$ are the oil variables. While the vinegar variables are allowed to be multiplied to any other variables, there are no oil times oil terms. Hence, if we guess for the vinegar variables we are left with a system of m linear equations in m variables. This has a high probability of being invertible (and one can always guess again for the vinegar variables if it is not). By composing with an affine transformation $\mathscr{T} : \mathbb{F}^n \to \mathbb{F}^n$ one gets the trapdoor function $\mathscr{P} = \mathscr{F} \circ \mathscr{T}$. This is indeed a trapdoor as by composing with $\mathscr{T}$, the oil and vinegar shape of the polynomials is lost and they appear just to be random. Thus for a oil and vinegar system the public key is $\mathscr{P}$ and the private key is $(\mathscr{F}, \mathscr{T})$. To sign a document Y, one first computes $\mathscr{F}^{-1}(Y) = Z$ by guessing the vinegar variables until $\mathscr{F}$ is an invertible linear system. Then one computes $\mathscr{T}^{-1}(Z) = W$. One verifies that W is a signature for Y by noting that $\mathscr{P}(W) = Y$.

Patarin originally proposed that the number of oil variables would equal the number of vinegar variables. Hence 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 [27]. This attack, however, is thwarted by making the number of vinegar variables sufficiently greater than the number of oil variables. Generally this is between 2 and 4 times as many vinegar variables to oil variables. Thus modern oil and vinegar schemes are called Unbalanced Oil and Vinegar (UOV) 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 [20]. UOV remains unbroken to this day, and offers competitive signing and verifying times compared to other signatures schemes. Its main flaw is its rather large key size. Thus there have been many modifications to UOV designed to reduce the key size. One, due to Petzoldt, is to use a pseudo-random number generator to generate large portions of the key from a smaller seed which is easier to store [34]. Other schemes use the basic mathematical structure of UOV, but modify it in a way to increase efficiency. However, any changes can generate weakness for the system as can be seen from the first round contender of the NIST competition HIMQ-3 [36] which was broken by the Singularity Attack from

Ding *et al.* [21]. Two of the nine signature schemes left in the second round of the competition are also based on UOV. Rainbow, originally proposed in 2005, gains efficiency by forming multiple UOV layers where the oil variables in the previous layers are the vinegar variables in the latter layers [20]. The other scheme first proposed in [3] is Lifted Unbalanced Oil and Vinegar (LUOV) whose core idea is to reduce its key size by selecting all the coefficients of its polynomials from $\mathbb{F}_2 = \{0, 1\}$. However, LUOV signs its messages in some extension field $\mathbb{F}_{2^r}$. LUOV was attacked by Ding *et al.* using the Subfield Differential Attack (SDA) in [22]. SDA uses the lifted form of the polynomials to always work in a smaller field and thus increase efficiency of direct attacks (those which try to solve the quadratic system outright) against LUOV. The authors of LUOV have amended their parameters in order to prevent SDA. However, in this paper we will show that LUOV is still vulnerable to a modified form of SDA which we will call the Nested Subset Differential Attack (NSDA).

1.4 Lifted Unbalanced Oil and Vinegar (LUOV)

The LUOV, proposed in [3], is a UOV scheme with three main modifications. Let $\mathbb{F}_{2^r}$ be an extension of $\mathbb{F}_2$, m and v be positive integers, and $n = m + v$. The central maps $\mathscr{F} : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^m$ is a system of quadratic maps $\mathscr{F}(X) = (F^{(1)}(X), \ldots, F^{(m)}(X))$ whose components are in oil and vinegar form

$$F^{(k)}(X) = \sum_{i=1}^{v}\sum_{j=i}^{n} \alpha_{i,j,k} x_i x_j + \sum_{i=1}^{n} \beta_{i,k} x_i + \gamma_k.$$

The first modification is that each $F^{(k)}$ is "lifted," meaning that the coefficients are taken from the prime field $\mathbb{F}_2$. Messages are still taken over the extension field, hence the name Lifted Unbalanced Oil Vinegar. The second modification is that the affine map $\mathscr{T}$ has the easier to store and computationally faster to sign form

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

This was first proposed by Czypek [11]. This does not affect security as for any given UOV private key $(\mathscr{F}', \mathscr{T}')$ there is highly likely an equivalent private key $(\mathscr{F}, \mathscr{T})$ where $\mathscr{T}$ is of the form above [41]. The third modification is that LUOV uses Petzdolt's method of generating the keys from a PRNG instead of storing them directly [34].

1.5 Our Contributions

In this paper we will first present the original SDA and then NSDA which is a modified version of the SDA attack which will defeat fully half of the new parameter sets used by LUOV. These parameters will fall well short of their targeted NIST security levels. We will also document an attack against one of these parameters sets which we were able to perform in under 210 min. Our attack does not rely on the oil and vinegar structure of LUOV, and can be seen as a way to solve "lifted" polynomial equations in general.

2 A Lemma on Random Maps

For both the Subfield Differential Attack and the Nested Subset Differential Attack we will require a short lemma on random maps which, under the assumption that quadratic systems of polynomials act like random maps, will allow us to say when it is possible to forge signatures.

Lemma 1. *Let A and B be two finite sets and $\mathscr{Q} : A \rightarrow B$ 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.

3 The Subfield Differential Attack

3.1 Transforming the Public Key into Better Form

In this section we recall the Subfield Differential Attack proposed in [22]. Let $\mathscr{P} : \mathbb{F}_{2^r}^n \rightarrow \mathbb{F}_{2^r}^m$ be a LUOV public key. Let $X = (x_1, \dots, x_n) \in \mathbb{F}_{2^r}^n$ be an indeterminate point. Then

$$\mathscr{P}(X) = \begin{cases} P^{(1)}(X) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,1} x_i x_j + \sum_{i=1}^{n} \beta_{i,1} x_i + \gamma_1 \\ P^{(2)}(X) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,2} x_i x_j + \sum_{i=1}^{n} \beta_{i,2} x_i + \gamma_2 \\ \vdots \\ P^{(m)}(X) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,m} x_i x_j + \sum_{i=1}^{n} \beta_{i,m} x_i + \gamma_m \end{cases}$$

where for each i, j, k we have $\alpha_{i,j,k}, \beta_{i,k}, \gamma_k \in \mathbb{F}_2$. Due to this special structure we are able to transform $\mathscr{P}$ to be over a subfield of $\mathbb{F}_{2^r}$ which, depending on the parameters, will allow us to forge signatures.

First we recall for every positive integer d which divides r we may represent $\mathbb{F}_{2^r}$ as a quotient ring

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

where $g(t)$ is a irreducible degree $s = r/d$ polynomial. For details see [28]. Let $\overline{X} = (\overline{x}_1, \ldots, \overline{x}_n) \in \mathbb{F}_{2^d}^n$ be an indeterminate point and $X' = (x'_1, \ldots, x'_n) \in \mathbb{F}_{2^r}^n$ be a random fixed point. So $\tilde{\mathscr{P}}(\overline{X}) := \mathscr{P}(\overline{X} + X') : \mathbb{F}_{2^d}^n \to \mathbb{F}_{2^r}^m$. Further this map is of a special form. Examining the kth component of $\tilde{\mathscr{P}}(\overline{X})$

$$\tilde{P}^{(k)}(\overline{X}) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k}(\overline{x}_i + x'_i)(\overline{x}_j + x'_j) + \sum_{i=1}^{n} \beta_{i,k}(\overline{x}_i + x'_i) + \gamma_k.$$

Expanding the above and separating the quadratic terms leads to

$$\begin{aligned}\tilde{P}^{(k)}(\overline{X}) =& \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k}(x'_i\overline{x}_i + x'_j\overline{x}_j + x'_ix'_j) \\ &+ \sum_{i=1}^{n} \beta_{i,k}(\overline{x}_i + x'_i) + \gamma_k + \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k}\overline{x}_i\overline{x}_j.\end{aligned}$$

We see that, due to $\alpha_{i,j,k} \in \mathbb{F}_2$, the coefficients of the quadratic terms $\overline{x}_i\overline{x}_j$ are all in the prime field. However, as the x'_i are random elements from $\mathbb{F}_{2^r}$, the coefficients of the linear $\overline{x}_i$ terms will contain all the powers of t up to $s - 1$. This means that, by grouping by the various powers of t, we may rewrite $\tilde{\mathscr{P}}(\overline{X})$ as

$$\tilde{\mathscr{P}}(\overline{X}) = \begin{cases} \tilde{P}^{(1)}(\overline{X}) = Q_1(\overline{X}) + \sum_{i=1}^{s-1} L_{i,1}(\overline{X})t^i \\ \tilde{P}^{(2)}(\overline{X}) = Q_2(\overline{X}) + \sum_{i=1}^{s-1} L_{i,2}(\overline{X})t^i \\ \vdots \\ \tilde{P}^{(m)}(\overline{X}) = Q_m(\overline{X}) + \sum_{i=1}^{s-1} L_{i,m}(\overline{X})t^i \end{cases}$$

3.2 Forging a Signature

Now suppose we wanted to forge a signature for a message Y. First decompose Y into the sum of vectors

$$Y = Y_0 + Y_1t + \cdots + Y_{s-1}t^{s-1}$$

where for each i, $Y_i = (y_{i,1}, \ldots, y_{i,m}) \in \mathbb{F}_{2^d}^m$.

First one finds the solution space S for the system of linear equations

$$A = \left\{ L_{i,j}(\overline{X}) = y_{i,j} : 1 \leq i \leq s-1, 1 \leq j \leq m \right\}.$$

As A is essentially a random system of linear equations, it will have a high probability to be full rank $(s-1)m$ (or n if $(s-1)m \geq n$). So the dimension of S will be

$$\dim(S) = \max\{n - (s-1)m, 0\}.$$

Next, one tries to solve the system of m quadratic equations

$$B = \left\{ Q_i(\overline{X}) = y_{0,i} : 1 \leq i \leq m, \overline{X} \in S \right\}.$$

If S is of large enough dimension, which depends on the choice of d, n, and m, The solution $\overline{X}$ to B yields $\tilde{P}(X) = Y$ which implies that $\mathscr{P}(\overline{X} + X') = Y$. Hence $\overline{X} + X'$ is the signature we seek. As the most costly step is solving the m quadratic equations of B over $\mathbb{F}_{2^d}$, we always choose d to be as small as possible for the S to likely have a solution according to Lemma 1 where in this case the domain is S and then range is $\mathbb{F}_{2^d}$. Generally, the domain will be much larger than the range for the attack and in this case we can assume that the probability for success on the first try is 1, or the domain is smaller and then the attack will fail as we almost never expect a solution to exist.

4 Nested Subset Differential Attack

4.1 The Change of Parameters for LUOV

In response to the Subfield Differential Attack, the authors of LUOV proposed the size of the extension r should be made prime so that the only subfield will be the prime field $\mathbb{F}_2$ [4]. They claim that given their new parameters, $\mathbb{F}_2^n$ will be far too small for a signature to exist for any given differential with any probability. The new parameters are in Table 1. We note that they are for different NIST security levels than before.

Table 1. The new parameter sets for LUOV

Name	Security level	(r, m, 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	$(79, 76, 341, 417)$

Indeed, by Lemma 1 the Subfield Differential Attack will not work without modification, but it is the claim of this paper that such a modification, which we will call the Nested Subset Differential Attack (NSDA), is indeed possible for the three cases for which $r = 7$. In fact for the level I security level the complexity will be brought into the range where the attack is not theoretical but possible in practice in under 210 min as we will later show. This is due to the special construction of lifted polynomials given by the following lemma.

4.2 A Lemma on Lifted Polynomials

Lemma 2. *Let*

$$\tilde{f}\left(X\right) = \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j}x_i x_j + \sum_{i=1}^{n} \beta_i x_i + \gamma$$

be a lifted polynomial and $A_0, A_1, \cdots, A_{\ell-1} \in \mathbb{F}_2^n$ *with*

$$A_i = \left(a_{i,1}, \cdots, a_{i,n}\right).$$

Set $\mathbf{A} = A_0 + A_1 t + A_2 t^2 + \cdots + A_{\ell-1}t^{\ell-1}$. *We have that for* $\tilde{f}\left(\mathbf{A} + Xt^{\ell}\right)$ *all the quadratic terms are coefficients of* $t^{2\ell}$*, the linear terms are coefficients of* $t^{\ell}, t^{\ell+1}, \cdots, t^{2\ell-1}$*, and the coefficients of* t^h *depends only on* $\alpha_{i,j}, \beta_i$*, and* A_k *for* $k \leq h$ *and* X *for* $h \geq \ell$*.*

Proof. This follows from the following calculation and the fact that for each $i, j \in \{1, \ldots, n\}$, $\alpha_{i,j}, \beta_i \in \mathbb{F}_2$.

$$\begin{aligned}
f(\mathbf{A} + Xt^{\ell}) = &\sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j} \left(\sum_{k=0}^{\ell-1} a_{k,i}t^k + x_i t^{\ell}\right)\left(\sum_{k=0}^{\ell-1} a_{k,j}t^k + x_j t^{\ell}\right) \\
&+ \sum_{i=1}^{n} \beta_i \left(\sum_{k=0}^{\ell-1} a_{k,i}t^k + x_i t^{\ell}\right) + \gamma \\
= &\sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j} \left(x_i x_j t^{2\ell} + x_i \sum_{k=0}^{\ell-1} a_{k,j}t^{k+\ell} + x_j \sum_{k=0}^{\ell-1} a_{k,i}t^{k+\ell}\right) \\
&+ \sum_{i=1}^{n} \beta_i x_i t^{\ell} + \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j} \sum_{h=0}^{2\ell-2} \left(\sum_{\substack{0 \leq k,k' \leq \ell \\ k+k'=h}} a_{k,i}a_{k',j}t^h\right) \\
&+ \sum_{i=1}^{n} \beta_i \left(\sum_{k=0}^{\ell} a_{k,i}t^k\right) + \gamma.
\end{aligned}$$

4.3 s-Truncation

It will also be convenient later to define the concept of s-truncation for an element of the extension field. For $0 \leq s \leq r-1$, we define the s-truncation of a element

$$a = \sum_{i=0}^{r-1} a_i t^i \quad \text{to be} \quad \overline{a}^s = \sum_{i=0}^{s} a_i t^i.$$

Similarly for a polynomial

$$f(\overline{X}) = \sum_{i=1}^{n} \sum_{j=i}^{n} a_{i,j} \overline{x_i x_j} + \sum_{i=1}^{n} b_i \overline{x_i} + c$$

we define the s-truncation to be term by term

$$\overline{f}^s(\overline{X}) = \sum_{i=1}^{n} \sum_{j=i}^{n} \overline{a_{i,j}}^s \overline{x_i x_j} + \sum_{i=1}^{n} \overline{b_i}^s \overline{x_i} + \overline{c}^s.$$

Finally, for a system of polynomials

$$\mathscr{G}(\overline{X}) = \left(g_1(\overline{X}), g_2(\overline{X}), \ldots, g_m(\overline{X})\right)$$

we define the s-truncation to by truncating each polynomial individually

$$\overline{\mathscr{G}}^s(\overline{X}) = \left(\overline{g_1}^s(\overline{X}), \overline{g_2}^s(\overline{X}), \ldots, \overline{g_m}^s(\overline{X})\right).$$

4.4 The Attack

Let $P : \mathbb{F}_{2^r}^n \to \mathbb{F}_{2^r}^m$ be a LUOV public key with $r = 7$ and suppose we want to forge a signature for a message $Y \in \mathbb{F}_{2^r}^m$. We will denote by $\overline{X} = (\overline{x}_1, \ldots, \overline{x}_n)$ an indeterminate in $\mathbb{F}_2^n$ and decompose the message Y into the sum of vectors

$$Y = Y_0 + Y_1 t + \cdots + Y_{r-1} t^{r-1}$$

where for each i, $Y_i = (y_{i,1}, \ldots, y_{i,m}) \in \mathbb{F}_2^m$.

Consider the set of polynomials in $\mathbb{F}_2[t]/\langle g(t) \rangle$ which are truncated to the third power

$$E := \left\{\overline{a}^3 : a \in \mathbb{F}_{2^r}\right\}.$$

Table 2 calculates the probability that there will exist a signature for Y in E^n for the relevant parameters using Lemma 1. In this case the domain is E^n which has a size of 2^{4n} and the range is $\mathbb{F}_{2^7}^m$ which has a size of 2^{7m}. So in each case the probability of success is $1 - \exp(-2^{4n}/2^{7m})$.

Table 2. Probability that a signature exists in E^n

Name	Probability
LUOV-7-57-197	$1 - \exp(-2^{617})$
LUOV-7-83-283	$1 - \exp(-2^{883})$
LUOV-7-110-374	$1 - \exp(-2^{2366})$

We thus see that it is very likely that we need to only consider signatures from E^n when we attempt to forge. Similar to SDA's usage of the differential

X' to transform the direct attack into solving equations over a subfield, we do not need to look over all of E^n at once but can instead construct a signature piece by piece using differentials. However, instead of choosing the differentials randomly, we will instead solve for them in such a manner that will eventually construct a signature. For our attack to be efficient, we will want to always solve no more than m quadratic equations over $\mathbb{F}_2$ with at least as many variables as equations. This can be done in four steps using Lemma 2.

First we see that

$$\overline{\mathscr{P}}^0(\overline{X}) = \begin{cases} Q_{0,1}(\overline{X}) \\ Q_{0,2}(\overline{X}) \\ \vdots \\ Q_{0,m}(\overline{X}) \end{cases}$$

where each $Q_{0,i}(\overline{X})$ is a quadratic polynomial over $\mathbb{F}_2$. So we may solve the system of m equations in n variables $\overline{\mathscr{P}}^0(\overline{X}) = Y_0$ using a direct attack method like exhaustive search [6], a variant of XL (eXtended Linerization) [10], or a Gröbner Basis method like F4 [24]. We will forestall discussion of which algorithm to use until Sect. 4.6. Let us call the solution we found A_0.

For the second step, let us examine $\overline{\mathscr{P}}^1(A_0 + \overline{X}t)$. By the definition of s-truncation, this will be a system of polynomials of degree at most 1 in t. Following from Lemma 2, the coefficients of the t^1 terms will be linear in the variables $\overline{X}$. Furthermore, the coefficients of the t^0 terms will depend only on A_0. As $\overline{\mathscr{P}}^0(A_0) = Y_0$, we see that

$$\overline{\mathscr{P}}^1(A_0 + \overline{X}t) = \begin{cases} y_{0,1} + L_{1,1}(\overline{X})t \\ y_{0,2} + L_{1,2}(\overline{X})t \\ \vdots \\ y_{0,m} + L_{1,m}(\overline{X})t \end{cases}$$

where each $L_{1,i}(\overline{X})$ is a linear polynomial over $\mathbb{F}_2$ in the variables $\overline{X}$. Now find a solution A_1 to the system of linear equations

$$\left\{ L_{1.i}(\overline{X}) = y_{1,i} : 1 \le i \le m \right\}.$$

Then we have $\overline{\mathscr{P}}^1(A_0 + A_1 t) = Y_0 + Y_1 t$.

For the third step, examine $\overline{\mathscr{P}}^2(A_0 + A_1 t + \overline{X}t^2)$. Again the s-truncation will make this a system of polynomials of degree 2 in t. Lemma 2 states that the coefficients of the t^2 terms will be linear in the variables $\overline{X}$. The coefficients of the t^0 terms will depend only on A_0, and the coefficients of the t^1 will depend only on A_0 and A_1. But by construction of A_0 and A_1 we see that

$$\overline{\mathscr{P}}^2(A_0 + A_1 t + \overline{X}t^2) = \begin{cases} y_{0,1} + y_{1,1}t + L_{2,1}(\overline{X})t^2 \\ y_{0,2} + y_{1,2}t + L_{2,2}(\overline{X})t^2 \\ \vdots \\ y_{0,m} + y_{1,m}t + L_{2,m}(\overline{X})t^2 \end{cases}$$

where each $L_{2,i}(\overline{X})$ is a linear polynomial over $\mathbb{F}_2$ in the variables $\overline{X}$. Again find a solution A_2 to the system of linear equations

$$\left\{L_{2.i}(\overline{X}) = y_{2,i} : 1 \le i \le m\right\}.$$

Then we have $\overline{\mathscr{P}}^2(A_0 + A_1 t + A_2 t^2) = Y_0 + Y_1 t + Y_2 t^2$.

As a final step, we drop the need for s-truncation and look at $\mathscr{P}(A_0 + A_1 t + A_2 t^2 + \overline{X}t^3)$. We note that this will be a system of polynomials of degree 6 in t, the highest degree for polynomials in $\mathbb{F}_2[t]/\langle g(t)\rangle$ as $r = 7$. Further, by Lemma 2, only the coefficients of the t^6 terms will be quadratic in $\overline{X}$. The coefficients of the t^3, t^4 and t^5 terms will be linear in $\overline{X}$. Finally, the coefficients of the t^0, t^1, t^2 terms depend only on A_0, A_0 and A_1, and A_0 A_1 and A_2 respectively. Let $\mathbf{A} = A_0 + A_1 t + A_2 t^2$. By construction of A_0, A_1, and A_2 we see that

$$\mathscr{P}(\mathbf{A} + \overline{X}t^3) = \begin{cases} y_{0,1} + y_{1,1}t + y_{2,1}t^2 + L_{3,1}(\overline{X})t^3 + L_{4,1}(\overline{X})t^4 \\ \qquad + L_{5,1}(\overline{X})t^5 + Q_{6,1}(\overline{X})t^6 \\ y_{0,2} + y_{1,2}t + y_{2,2}t^2 + L_{3,2}(\overline{X})t^3 + L_{4,2}(\overline{X})t^4 \\ \qquad + L_{5,2}(\overline{X})t^5 + Q_{6,2}(\overline{X})t^6 \\ \vdots \\ y_{0,m} + y_{1,m}t + y_{2,m}t^2 + L_{3,m}(\overline{X})t^3 + L_{4,m}(\overline{X})t^4 \\ \qquad + L_{5,m}(\overline{X})t^5 + Q_{6,m}(\overline{X})t^6 \end{cases}$$

Now we proceed largely in the same manner as the last step in the SDA attack. Find the solution space S for the system of linear equations

$$A = \left\{L_{i,j}(\overline{X}) = y_{i,j} : 3 \le i \le 5, 1 \le j \le m\right\}.$$

As A will most likely be full rank $3m$, the dimension of S will have high probability of being $n - 3m$. Thus, the system of m quadratic equations

$$B = \left\{Q_{6,j}(\overline{X}) = y_{6,j} : 1 \le j \le m, \overline{X} \in S\right\}$$

has a high probability of having a solution given the parameter sets of LUOV which we record in Table 3. Again, we used Lemma 1 with the domain being S which has size 2^{n-3m}, and the range being $\mathbb{F}_2^m$ which has size 2^m. So the probability of success is $1 - \exp(-2^{n-4m})$.

Table 3. Probability of success for NSDA

Name	Probability
LUOV-7-57-197	$1 - \exp(-2^{26})$
LUOV-7-83-283	$1 - \exp(-2^{34})$
LUOV-7-110-374	$1 - \exp(-2^{344})$

Find a solution A_3 to B. Then we see that

$$\mathscr{P}(A_0 + A_1 t + A_2 t^2 + A_3 t^3) = Y$$

and thus $\sigma = A_0 + A_1 t + A_2 t^2 + A_3 t^3$ is a forged signature for Y.

Note that in each case we assumed that it was possible to find the solutions A_i for the various systems. The last quadratic system is when this is most unlikely, and still we see that the odds are overwhelmingly in our favor for the parameter sets we attacked for the solutions to exist assuming that polynomial systems act as random maps. Thus, we may ignore the potential that a solution does not exist in our attack for any step, and even if that were the case one can always go back a previous step for a different solution and try again.

For the different parameter sets this is no longer so. They use a larger value for r, which means that the number of linear equations to solve along side the final quadratic system also increases to the point where we no longer expect a final solution to exist. This bring into question when LUOV is safe from SDA and NSDA, which depends on the relationship between n, m, r, and any factors d of r, but is still competitive. This is beyond the scope of this paper, and further work will need to be done to see the exact value of the lifting modification.

4.5 Hiding the Signature

It might be argued that signatures that come from E^n are in a very special shape and thus can be rejected as obviously forged. However, it is possible to hide the shape of the signatures generated from the NSDA attack. Due to the special shape of the lifted polynomials, it is possible to know about preimages of a more generic form which are connected to the preimages we can find. Let $\mathscr{P}$ be a LUOV public key so that

$$\mathscr{P}(X) = \begin{cases} P^{(1)}(X) = \sum_{i=1}^{n} \sum_{j=i}^{n} \alpha_{i,j,1} x_i x_j + \sum_{i=1}^{n} \beta_{i,1} x_i + \gamma_1 \\ P^{(2)}(X) = \sum_{i=1}^{n} \sum_{j=i}^{n} \alpha_{i,j,2} x_i x_j + \sum_{i=1}^{n} \beta_{i,2} x_i + \gamma_2 \\ \\ P^{(m)}(X) = \sum_{i=1}^{n} \sum_{j=i}^{n} \alpha_{i,j,m} x_i x_j + \sum_{i=1}^{n} \beta_{i,m} x_i + \gamma_m \end{cases}$$

Suppose we wanted to forge a signature for a message $Y = (y_1, \dots, y_m) \in \mathbb{F}_{2^r}^m$. As we are in a finite field of characteristic 2, we may take square roots of any element. For some natural number N, define a vector $Z = (z_1, \dots, z_m) = Y^{1/2^N}$ by which we mean that, for each i, $z_i = y_i^{1/2^N}$ the 2^Nth root of y_i. Now let $X = (x_1, \dots, x_n) \in E^n$ be a signature for Z so that $\mathscr{P}(X) = Z$. Define $X^{2^N} = (x_i^{2^N}, \dots, x_n^{2^N})$. Let us recall the freshman's dream.

Theorem 1 (Freshman's Dream). *If $\mathbb{F}$ is a field of characteristic p then for any natural number N and elements $x, y \in \mathbb{F}$ we have $(x+y)^{p^N} = x^{p^N} + x^{p^N}$.*

Then examining the kth component of $\mathscr{P}(X^{2^N})$ we see that due to the freshman's dream and the fact that the coefficients of $\mathscr{P}$ are in $\mathbb{F}_2$

$$\begin{aligned} P^{(k)}(X^{2^N}) &= \sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k} x_i^{2^N} x_j^{2^N} + \sum_{i=1}^{n} \beta_{i,k} x_i^{2^N} + \gamma_k \\ &= \left(\sum_{i=1}^{n}\sum_{j=i}^{n} \alpha_{i,j,k} x_i x_j + \sum_{i=1}^{n} \beta_{i,k} x_i + \gamma_1 \right)^{2^N} \\ &= z_k^{2^N} = y_k. \end{aligned}$$

As the elements of X are degree three polynomials in $\mathbb{F}_2[t]/\langle g(t) \rangle$, X^{2^N}'s elements will appear to be generic degree six polynomials. Now, the signature can still be seen by checking the 2^Nth roots for each N less than r, but this procedure still masks the forged signature from against lazy implementations of the verification process.

4.6 Complexity

The complexity of our attack is determined by solving the two quadratic systems of m equations over $\mathbb{F}_2$. The overhead from solving the linear systems we may ignore as the size of the linear systems is always not much larger than the quadratic systems, and linear systems are much more efficient to solve.

Let us take a system $\mathscr{P} = \left(P^{(1)}(X), \dots, P^{(m)}(X)\right)$ of m quadratic equations in n variables over $\mathbb{F}_2$ and attempt to find a solution. The best method in our case given the small field size and the limited number of variables we will have is exhaustive search. In our practical experiment on LUOV-7-57-197, we used a variant of the "forcepq_fpga" algorithm [5,6], so this algorithm is how we will estimate the complexity of solving the system. We will give a brief sketch of the main idea here.

We will denote the solution set of the first k equations as

$$Z_\ell = \{A \in \mathbb{F}_2 \mid P^{(i)}(A) = 0, 1 \leq i \leq \ell\}.$$

For some well chosen ℓ, the algorithm first utilizes Grey-code and partial derivatives to find Z_ℓ by solving the first ℓ equations individually. We begin by ordering the elements of $\mathbb{F}_2$ according to a Grey-code order $A_1, A_2, \ldots, A_{2^n}$. This means that for an element $A_s \in \mathbb{F}_2^n$, A_{s+1} will only have one component different than A_s. The authors of [5] noticed that, as we are working under $\mathbb{F}_2$ and if A_{s+1} differs from A_s only at the ith component

$$P^{(k)}(A_{s+1}) = P^{(k)}(A_s) + \frac{\partial P^{(k)}}{\partial x_i}(A_{s+1}).$$

As the partial derivative is one degree smaller, it is more efficient to evaluate. It was also found that this trick can be used recursively for evaluating the first partial derivatives utilizing the second partial derivatives.

Notice, though, that Z_ℓ is no longer in Gray-code order as it is essentially a random subset of $\mathbb{F}_2^n$. Thus, it is not possible to fully utilize the Gray-code method to compute $Z_{\ell+1}$ from Z_ℓ. One would have to add multiple evaluations of different partial derivatives, one for each change in component, when selecting the next element of Z_ℓ. This was only found to be twice as efficient as simply evaluating the original equations in view of finding Z_m at the very end.

It was estimated in [5] that the number of bit operations for finding all the solutions would be $\log_2(n)2^{n+2}$ for a determined system ($n = m$) with an optimal value of $\ell = 1 + \log_2(n)$. We will use this estimate on determined systems as for the cases we consider we will have more variables than equations. As we only need one solution we can randomly assign values until the system is either determined or only slightly underdetermined ($n > m$) if we want a solution on the first attempt. In our experiment we guessed for all but $m+2$ of the variables to assure a solution first try, so we will do likewise in our estimate.

We will note though that if n is multiple times the size of m, we can first use the method of Thomae and Wolf [40], which is an improvement of the work of the Kipnis, Patarin, and Goubin [26], to reduce the number of variables *and* equations. While we will not go into the details of the method in this paper, the core idea is to make the random system act as if that is was at least partly an oil and vinegar system. By this we mean we attempt to find some linear transformation of the variables $\mathscr{S}$ such that $\mathscr{P} \circ \mathscr{S}$ has a set of vinegar variables V and a set of oil variables O. The result is part of the resulting system is linear in the oil variables after fixing the vinegar variables. As we are in characteristic 2, square terms act linearly. Thus, we search for $\mathscr{S}$ to set each $O \times O$ coefficient $\alpha_{i,j,k} = 0$ for $i \neq j$. Thomae and Wolf showed that this process can be done solving a relatively small system of linear equations. The statement of their result is as follows.

Theorem 2 (Thomae and Wolf). *By a linear change of variables, the complexity of solving an under-determined 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. Furthermore, provided $\lfloor\omega\rfloor | m$ the complexity can be further reduced to the complexity of solving a determined quadratic system of $m - \lfloor\omega\rfloor$ equations [40].*

In Table 4 we compute the complexity for solving the final quadratic system. This will be the most complex part of the attack as we had to first solve a linear system. We will have approximately $n - 3m$ variables and m equations. We note that as $(n - 3m)/m < 2$ in each case, Thomae and Wolf's method will not apply. We will guess all but $m + 2$ variables and estimate the complexity as $\log_2(m + 2)2^{m+4}$.

Table 4. Complexity in terms of number of bit operations

Name	$\log_2$ NSDA's complexity (NIST Requirement)
LUOV-7-57-197	61 (143)
LUOV-7-83-283	89 (207)
LUOV-7-110-374	116 (272)

As the classical $\log_2$ classical gate operations for NIST security level I is 143, III is 207, and V is 272 [31], we see that LUOV falls short in every category for these parameters. Moreover, the actual complexity for NSDA is possible in practice as we show with experimental results in Sect. 4.7.

Before we continue, we will mention that if the subfield over which we solved had been larger, or if the number of variables to guess for had been too great, then exhaustive search would not be the optimal choice for the solver for the quadratic systems. Generally, after applying the method of Thomae and Wolf, either XL [10] with the Block Wiedemann Algorithm [9] or the F4 algorithm by Faugère [24] is the preferred choice for such systems using a hybrid method [2] (meaning guessing a certain number of variables before applying the mentioned algorithms). The complexity of both algorithms relies on solving/reducing very large, sparse Macaulay matrices. Roughly, the highest degree found in XL is denoted by D_0 (called the operating degree), and the highest degree in F4 is D_{reg} (called the degree of regularity [14,15,19,23]). Yeh *et al.* [42] have shown that for the resulting overdetermined systems after using the hybrid method, $0 \leq D_0 - D_{reg} \leq 1$ and often $D_0 = D_{reg}$. So the matrices are roughly the same size, but XL is sparser and is thus the preferred method to use. Please see [42] for full details.

4.7 Experimental Results

We have performed practical experiments on the LUOV parameter set LUOV-7-57-197.

For the hardware, we used a field-programmable gates array cluster from Sciengines, a "Rivyera S6-LX150" with 64 Xilinx Spartan 6 LX150 FPGAs chips. The LX150 were so named because each contains nearly "150,000 gate equivalent units". They were driven on 8 PCI express cards in a chassis containing a Supermicro motherboard, an Intel Xeon(R) CPU (E3-1230 V2). When new in 2012, the machine cost 55,000 EUR. Although not directly comparable, a machine with current FPGAs costing the same 55,000 EUR today will probably have at least 2× as much computing power and cost less in electricity.

We use a variant of the "forcepq_fpga" algorithm from the paper [6], using the input format of the Fukuoka MQ Challenge. We processed the early parts of our LUOV attack using the computer algebra system Magma and output the resulting system in this format, which is basically binary quadratic systems with zero-one coefficients lined up in graded reverse lexicographic order.

The "forcemp_fpga" implementation allows us to test 2^{10} input vectors per cycle (at 200 MHz) per FPGA chip. In general this lets us solve a 48×48 MQ system in a maximum of slightly less than 23 min using one single chip, or find a solution to $n \times m$ quadratic equations, where $n \geq m$, in $2^{m-48} \times 23$ min. We could accelerate this somewhat if we can implement a variation of the Joux-Vitse algorithm.

For a 55-equation system, using all 64 FPGAs, the maximum is 46 min. In general it is a little shorter. The expectation is half of that or 23 min. For a 57-equation system, it is 4 times that, hence about 3 h, expectation is about half of that or 92 min. When we solved the 59-variable, 57-equation system in practice, the run ended after 105 min. This, like all our runs in this experiment, happened to be slightly unlucky.

As there are two quadratic systems to solve, we can forge a signature in under 210 min.

5 Inapplicability to Non-lifted Schemes

Before we conclude, lets discuss why NSDA or any similar attack does not work on UOV [33], Rainbow [18], or any other multivariate scheme which does not use the lifting modification. In these schemes, though some coefficients in the central map are forced to be 0 (like the oil × oil coefficients in UOV and Rainbow) to allow efficient pre-image finding, most of the coefficients in the central maps are taken randomly from a finite field $\mathbb{F}_q$. Thus, in the public key $\mathscr{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ all of the coefficients are seemingly random elements of $\mathbb{F}_q$. This makes any differential we add seemingly mixed randomly.

To be explicit, Let us assume that $\mathbb{F}_q$ contains a subfield $\mathbb{F}_{q'}$ so that $\mathbb{F}_q \cong \mathbb{F}_{q'}[t]/\langle g(t) \rangle$ where $\deg(g) = s$. We will assume that $\mathbb{F}_{q'}$ is to small to find pre-images in. Let $\overline{X} = (\overline{x}_1, \ldots, \overline{x}_n)$ be an indeterminate point in $\mathbb{F}_{q'}^n$, $f(t) \in \mathbb{F}_{q'}[t]$ (say $f(t) = t$ like in NSDA), and $A = (a_1, \ldots, a_n)$ be a fixed point (whether in a special form like in NSDA or not). Let $\tilde{\mathscr{P}}(\overline{X}) := \mathscr{P}(A + \overline{X} f(t))$. Similar to the SDA section we find that in the kth component of $\tilde{\mathscr{P}}$

$$\begin{aligned}\tilde{P}^{(k)}(\overline{X}) = &\sum_{i=1}^{n}\sum_{j=i}^{n}\alpha_{i,j,k}(a_j\overline{x}_i f(t) + a_i\overline{x}_j f(t) + a_i a_j)\\ &+ \sum_{i=1}^{n}\beta_{i,k}(\overline{x}_i f(t) + a_i) + \gamma_k + \sum_{i=1}^{n}\sum_{j=i}^{n}\alpha_{i,j,k}\overline{x}_i\overline{x}_j f(t)^2.\end{aligned}$$

Note that there are no restrictions on the coefficients, $\alpha_{i,j,k}, \beta_{i,k}$ and γ_k as they are random elements from $\mathbb{F}_{q^r}$. The quadratic terms' coefficients will contain powers of t from t^0 to t^{s-1}. Hence, we are trading one random quadratic system $\mathscr{P}$ which $\mathbb{F}_{q'}^n$ is too small to find pre-images in for another equally random quadratic system $\tilde{\mathscr{P}}$ which $\mathbb{F}_{q'}^n$ is still too small. So, NSDA is inapplicable to non-lifted systems.

6 Conclusion

We have proposed a modified version of the Subfield Differential Attack called Nested Subset Differential Attack which fully breaks half the parameters set forward by the round 2 version of Lifted Unbalanced Oil and Vinegar. We reduced attacking these parameters sets to the problem of solving quadratic equations over the prime field $\mathbb{F}_2$. This makes our attack effective enough to be performed practically. As our attack did not use the Unbalanced Oil and Vinegar Structure of LUOV, it can be seen as a method of solving lifted quadratic systems in general. We feel that more research into solving these types of quadratic systems using the NSDA attack is needed. We also performed experimental attacks on actual LUOV parameters and were able to forge a signature in under 210 min.

References

1. Bernstein, D.J.: Introduction to post-quantum cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_1
2. Bettale, L., Faugère, J.-C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. J. Math. Cryptol. **3**(3), 177–197 (2009)
3. Beullens, W., Preneel, B.: Field lifting for smaller UOV public keys. In: Patra, A., Smart, N.P. (eds.) INDOCRYPT 2017. LNCS, vol. 10698, pp. 227–246. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71667-1_12
4. Beullens, W., Preneel, B., Szepieniec, A., Vercauteren, F.: Luov signature scheme proposal for NIST PQC project (round 2 version) (2019)
5. Bouillaguet, C., et al.: Fast exhaustive search for polynomial systems in $\mathbb{F}_2$. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 203–218. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15031-9_14

6. Bouillaguet, C., Cheng, C.-M., Chou, T., Niederhagen, R., Yang, B.-Y.: Fast exhaustive search for quadratic systems in $\mathbb{F}_2$ on FPGAs. In: Lange, T., Lauter, K., Lisoněk, P. (eds.) SAC 2013. LNCS, vol. 8282, pp. 205–222. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43414-7_11
7. Buchmann, J., Ding, J. (eds.): Post-Quantum Cryptography. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88702-7
8. Campagna, M., et al.: ETSI whitepaper: Quantum safe cryptography and security. Technical report, ETSI (2015)
9. Coppersmith, D.: Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. Math. Comput. **62**(205), 333–350 (1994)
10. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_27
11. Czypek, P.: Implementing multivariate quadratic public key signature schemes on embedded devices. Ph.D. thesis, Citeseer (2012)
12. Diffie, W., Hellman, M.: New directions in cryptography. IEEE Trans. Inf. Theory **22**(6), 644–654 (1976)
13. Ding, J., Gower, J.E., Schmidt, D.: Multivariate Public Key Cryptosystems. Advances in Information Security, vol. 25. Springer, Cham (2006). https://doi.org/10.1007/978-0-387-36946-4
14. Ding, J., Hodges, T.J.: Inverting HFE systems is quasi-polynomial for all fields. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 724–742. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_41
15. Ding, J., Kleinjung, T.: Degree of regularity for HFE-. IACR Cryptology ePrint Archive 2011, 570 (2011)
16. Ding, J., Petzoldt, A.: Current state of multivariate cryptography. IEEE Secur. Privacy **15**(4), 28–36 (2017)
17. Ding, J., Petzoldt, A., Wang, L.: The cubic simple matrix encryption scheme. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 76–87. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_5
18. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005). https://doi.org/10.1007/11496137_12
19. Ding, J., Yang, B.-Y.: Degree of regularity for HFEv and HFEv-. In: Gaborit, P. (ed.) PQCrypto 2013. LNCS, vol. 7932, pp. 52–66. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38616-9_4
20. Ding, J., Yang, B.-Y., Chen, C.-H.O., Chen, M.-S., Cheng, C.-M.: New differential-algebraic attacks and reparametrization of rainbow. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 242–257. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_15
21. Ding, J., Zhang, Z., Deaton, J.: The singularity attack to the multivariate signature scheme HiMQ-3. Advances in Mathematics of Communications (2019)
22. Ding, J., Zhang, Z., Deaton, J., Schmidt, K., Vishakha, F.: New attacks on lifted unbalanced oil vinegar. In: The 2nd NIST PQC Standardization Conference (2019)
23. Dubois, V., Gama, N.: The degree of regularity of HFE systems. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 557–576. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17373-8_32
24. Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases (F4). J. Pure Appl. Algebra **139**(1–3), 61–88 (1999)

25. Johnson, D.S., Garey, M.R.: Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman, New York (1979)
26. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_15
27. Kipnis, A., Shamir, A.: Cryptanalysis of the oil and vinegar signature scheme. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 257–266. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055733
28. Lidl, R., Niederreiter, H.: Finite Fields, vol. 20. Cambridge University Press, Cambridge (1997)
29. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Barstow, D., et al. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-45961-8_39
30. National Institute of Standards and Technology: Workshop on cybersecurity in a post-quantum world. Technical report, National Institute of Standards and Technology (2015)
31. 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)
32. Patarin, J.: Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 248–261. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-44750-4_20
33. Patarin, J.: The oil and vinegar algorithm for signatures. In: Dagstuhl Workshop on Cryptography (1997)
34. Petzoldt, A., Bulygin, S.: Linear recurring sequences for the UOV key generation revisited. In: Kwon, T., Lee, M.-K., Kwon, D. (eds.) ICISC 2012. LNCS, vol. 7839, pp. 441–455. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37682-5_31
35. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
36. Shim, K.-A., Park, C.-M., Kim, T.: HiMQ-3: A high speed signature scheme based on multivariate quadratic equations, NIST submission (2017). Internet: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions
37. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999)
38. Tao, C., Diene, A., Tang, S., Ding, J.: Simple matrix scheme for encryption. In: Gaborit, P. (ed.) PQCrypto 2013. LNCS, vol. 7932, pp. 231–242. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38616-9_16
39. Tao, C., Xiang, H., Petzoldt, A., Ding, J.: Simple matrix - a multivariate public key cryptosystem (MPKC) for encryption. Finite Fields Appl. **35**, 352–368 (2015)
40. Thomae, E., Wolf, C.: Solving underdetermined systems of multivariate quadratic equations revisited. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) PKC 2012. LNCS, vol. 7293, pp. 156–171. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30057-8_10
41. Wolf, C., Preneel, B.: Equivalent keys in multivariate quadratic public key systems. J. Math. Cryptol. **4**(4), 375–415 (2011)
42. Yeh, J.Y.-C., Cheng, C.-M., Yang, B.-Y.: Operating degrees for XL vs. F_4/F_5 for generic $\mathscr{MQ}$ with number of equations linear in that of variables. In: Fischlin, M., Katzenbeisser, S. (eds.) Number Theory and Cryptography. LNCS, vol. 8260, pp. 19–33. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42001-6_3