Field lifting for smaller UOV public keys

Ward Beullens and Bart Preneel

imec-COSIC KU Leuven, Belgium,
ward.beullens@esat.kuleuven.be,
bart.preneel@esat.kuleuven.be

Abstract. Most Multivariate Quadratic (MQ) signature schemes have a very large public key, which makes them unsuitable for many applications, despite attractive features such as speed and small signature sizes. In this paper we introduce a modification of the Unbalanced Oil and Vinegar (UOV) signature scheme that has public keys which are an order of magnitude smaller than other MQ signature schemes. The main idea is to choose UOV keys over the smallest field $\mathbb{F}_2$ in order to achieve small keys, but to lift the keys to a large extension field, where solving the MQ problem is harder. The resulting Lifted UOV signature scheme is very competitive with other post-quantum signature schemes in terms of key sizes, signature sizes and speed.

Keywords: Post-Quantum Cryptography, Multivariate Cryptography, Signature Schemes, Unbalanced Oil and Vinegar, Key Size Reduction

1 Introduction

When large scale quantum computers are built, they will be able to break nearly all public key cryptography that is being used today, including RSA [25], DSA [17] and ECC. This is because these schemes rely on the hardness of number theoretic problems such as integer factorization and finding discrete logarithms, which can be solved efficiently by Shor's Algorithm [26]. Even if it would take 10 or 20 years to build large scale quantum computers, upgrading our current systems may be very slow and some stored data requires long term protection (in particular for confidentiality). To avert a potential catastrophe, post-quantum cryptography should be designed, implemented and deployed well before large scale quantum computers are built.

During recent years, the research on post-quantum cryptography has been accelerating. One of the goals of the EU-funded PQCRYPTO project is to develop and standardize post-quantum algorithms [1]. Recently NIST, the US National Institute for Standards and Technology, has started the process of selecting post-quantum algorithms for standardization [19]. According to both PQCRYPTO and NIST, multivariate cryptography is one of the major candidates for providing post-quantum security. Multivariate cryptography is based on the hardness of some problems related to multivariate polynomials over finite fields, such as solving multivariate polynomial equations. In general, multivariate cryptography

is very fast and requires only moderate computational resources, which makes it attractive for applications in low-cost devices. However, a disadvantage of multivariate cryptography is its large public keys, which can be prohibitive for many applications. Some work in mitigating this problem in the case of the UOV and Rainbow signature schemes has been published by Petzoldt [22], who managed to reduce the key size by a factor of 8 in the case of UOV and a factor of 3 in the case of the Rainbow signature scheme. His proposal makes a small modification to the key generation algorithm and exploits the fact that a large part of the public key can be freely chosen by the user. One can then choose to generate this part using a Pseudo-Random Number Generator (PRNG), and to only store the seed for the PRNG. In this paper we introduce a new idea to reduce the size of the public keys of UOV dramatically, by lifting the public and central maps to an extension field. The new idea is compatible with the ideas of Petzoldt and together they provide public keys that are up to 10 times smaller than if we were to use only Petzoldt's modification of UOV.

Before introducing the Lifted UOV signature scheme in Sect. 5, we present an overview of the MQ problem in Sect. 2 and the UOV signature and how it was improved by Pezoldt in Sects. 3 and 4. We finish with a brief description of our software implementation in Sect. 6 and conclude in Sect. 7.

2 The MQ problem

The security of an MQ signature scheme relies on the hardness of the MQ-problem. We give a brief discussion of the problem here.

MQ Problem. Given a quadratic polynomial map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ over a finite field $\mathbb{F}_q$, find $\mathbf{x} \in \mathbb{F}_q^n$ that satisfies $\mathcal{P}(\mathbf{x}) = \mathbf{0}$.

It is known that the MQ problem is NP-hard [18]. Therefore it is unlikely that there are (quantum) algorithms that solve the hardest instances of the MQ problem in polynomial time. The problem is also believed to be hard on average in the case $n \approx m$. Only exponential time algorithms are known to solve random instances of the problem for these parameters.

Systems with $n = m$ are called determined systems; these are the most difficult systems to solve. When $n < m$ a system is called overdetermined, and when $n > m$ the system is called underdetermined. Thomae et al. showed that finding a solution for an underdetermined system with $n = \alpha m$ can be reduced to finding a solution of a determined system with only $m+1-\lfloor \alpha \rfloor$ equations [27]. This means that as a system becomes more underdetermined it becomes easier to solve. This fact will become important in the security analysis of UOV.

2.1 Classical algorithms

The best known classical algorithms to solve the MQ-problem for generic determined systems over finite fields use the hybrid approach [5, 6]. This approach

combines exhaustive search with Gröbner basis computations. In this approach k variables are fixed to random values and the remaining $n-k$ variables are found with a Gröbner basis algorithm such as F_4, F_5 or XL. If no assignment to the remaining $n-k$ variables exists that solves the system, the procedure starts again with a different guess for the first k variables. We require on average roughly q^k Gröbner basis computations until a solution is found. As a result, the optimal value of k decreases as q increases. The complexity of computing a Gröbner basis for a system of polynomials depends critically on the degree of regularity (d_{reg}) of that system. Though it is of little importance to the rest of the paper, we refer to Bardet [2] for a precise definition of the degree of regularity. The complexity of the F_5 algorithm is given by

$$C_{F_5}(n, d_{reg}) = O\left(\binom{n + d_{reg}}{d_{reg}}^{\omega}\right),$$

where $2 \leq \omega < 3$ is the constant in the complexity of matrix multiplication. Therefore the complexity of the hybrid approach is

$$C_{\text{Hybrid}F_5(n,d_{reg},k)} = O\left(q^k \binom{n - k + d_{reg}(k)}{d_{reg}(k)}^{\omega}\right), \tag{1}$$

where $d_{reg}(k)$ stand for the degree of regularity of the system after fixing the values of k variables.

Determining the degree of regularity for a specific polynomial system is difficult, but for a certain class of systems, called semi-regular systems, it is known that the degree of regularity can be deduced from the number of equations m and the number of variables n [2, 8]. In particular, for quadratic semi-regular systems the degree of regularity is the degree of the first term in the power series of

$$S_{m,n}(x) = \frac{(1-x^2)^m}{(1-x)^n}$$

with a non-positive coefficient. This gives a practical method to calculate the degree of regularity of any semi-regular system. Empirically, polynomial systems that are randomly chosen have a very large probability of being semi-regular and it is conjectured that most systems are semi-regular systems. For the definition and the theory of semi-regular systems we refer to chapter 3 of the PhD thesis of Bardet [2].

2.2 Quantum algorithms

Currently, there are no specialized quantum algorithms that solve polynomial systems over finite fields. However, Grover's algorithm [13] can be used to speed up the brute force part of the hybrid approach. This approach gives a quadratic speedup for the brute force part of the attack, so the new complexity would be

$$C_{\text{Hybrid}F_5(n,d_{reg},k)} = O\left(q^{k/2} \binom{n - k + d_{reg}(k)}{d_{reg}(k)}^{\omega}\right), \tag{2}$$

where the difference with (1) is that we have the factor $q^{k/2}$ instead of q^k. However it should be noted that this approach requires sequentially running $q^{k/2}$ Gröbner basis computations on a quantum computer. This would be an incredible feat because even for moderately sized polynomial systems this would require gigabytes worth of qubits and days of computation without decoherence. Also, note that the gains of parallelizing Grover search grow only with the square root of the number of independent computers used, instead of a linear growth for the classical brute force search [28]. Nevertheless, in the security analysis of the signature scheme proposed in this paper we will be cautious and assume that these kinds of attacks on the MQ problem are possible and we will make our parameter choices accordingly. This has the additional benefit of providing a large safety margin against classical attacks.

Remark 1. Typically the optimal value of k, i.e. the number of variables that is guessed by brute force, is quite small (eg. 2,3 or 4), this does *not* mean that the hybrid approach is only a marginal improvement over a direct Gröbner basis computation. Guessing only a few variables can drastically reduce the degree of regularity of a system. For example, guessing only one variable in a determined semi-regular system of polynomials roughly reduces the degree of regularity by half! The idea of lifting a public key to an extension field is a countermeasure to the hybrid approach. By working in a large extension field (eg. $\mathbb{F}_{2^{64}}$) we ensure that guessing even a single variable is computationally too expensive.

3 The UOV signature scheme

The UOV or Unbalanced Oil and Vinegar digital signature scheme is a multivariate quadratic (MQ) signature scheme. It is a slightly modified version of the original Oil and Vinegar signature scheme that was proposed by Patarin in 1997 [20]. With the right parameter choices UOV has withstood all cryptanalysis since 1997 and it is one of the best studied and most promising MQ signature schemes.

3.1 Description of UOV

The UOV signature scheme uses a one-way function $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, which is a multivariate quadratic polynomial map over some finite field $\mathbb{F}_q$. The trapdoor is a factorization $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$, where $\mathcal{T} : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is an invertible linear map, and $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is a quadratic map whose components $f_1, \cdots, f_m$ are of 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 $v = n - m$. We say that the first v variables $x_1, \cdots, x_v$ are the vinegar variables, whereas the remaining m variables are the oil variables. The components of $\mathcal{F}$ are quadratic polynomials in the variables x_i such that there are no

quadratic terms which contain two oil variables. One could say that the vinegar variables and the oil variables are not fully mixed, which is where their names come from. [1]

How does the trapdoor $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ help to invert the function $\mathcal{P}$? Given a target $\mathbf{x} \in \mathbb{F}_q^m$ a solution $\mathbf{y}$ for $\mathcal{P}(\mathbf{y}) = \mathbf{x}$ can be found by first solving $\mathcal{F}(\mathbf{y}') = \mathbf{x}$ for $\mathbf{y}'$ and then computing $\mathbf{y} = \mathcal{T}^{-1}(\mathbf{y}')$. The system $\mathcal{F}(\mathbf{y}') = \mathbf{x}$ can be solved efficiently by randomly choosing the values of the vinegar variables. If we substitute these values in the equations the remaining system only contains linear equations, because every quadratic term contains at least one vinegar variable and thus turns into a linear or constant term after substitution. The remaining linear system can be solved using linear algebra. In the event that there are no solutions we can simply try again with a different choice for the vinegar variables.

The trapdoor function is then combined with a collision resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$ into a signature scheme using the standard hash-and-sign paradigm. The resulting key generation, signature generation and verification algorithms of the UOV signature scheme are described in Algorithms 1, 2 and 3.

Algorithm UOVGenerateKeys

input: Random bits to generate $\mathcal{F}$ and $\mathcal{T}$
output: $\mathcal{P}$ — A public key
$(\mathcal{F}, \mathcal{T})$ — A corresponding secret key

1: $\mathcal{F} \leftarrow$ A randomly chosen UOV system
2: $\mathcal{T} \leftarrow$ A randomly chosen linear map $\mathbb{F}_q^n \to \mathbb{F}_q^n$
3: $\mathcal{P} \leftarrow \mathcal{F} \circ \mathcal{T}$
4: **return** $\mathcal{P}$ and $(\mathcal{F}, \mathcal{T})$

Algorithm 1. The UOV key pair generation algorithm

3.2 Attacks against UOV

Direct attack. This attack tries to forge a signature s for a message M by solving the polynomial system $\mathcal{P}(s) = \mathcal{H}(M)$. An attacker can use the trick of

[1] However it is not a very good name because in reality oil mixes with oil and vinegar mixes with vinegar but no mixing happens between oil and vinegar, and this is not what happens in UOV polynomials. A better name would have been hen variables and rooster variables because hens can get along with hens and roosters, but two roosters start a fight when they appear in the same term. Moreover, this foreshadows the fact that in order for the signature scheme to be secure, the number of hen (vinegar) variables should be larger than the number of rooster (oil) variables. Nevertheless, we will stick to the traditional naming of oil and vinegar variables.

Algorithm UOVSign

input: $(\mathcal{F}, \mathcal{T})$ — A secret key
M — A message to sign
output: $\mathbf{s}$ — A signature for the message M

$\mathbf{h} \leftarrow \mathcal{H}(M)$
while No solution $\mathbf{s}'$ to the system $\mathcal{F}(\mathbf{s}') = \mathbf{h}$ is found **do**
  Assign random values to the first v entries of $\mathbf{s}'$
  Substitute these values into $\mathcal{F}(\mathbf{s}') = \mathbf{h}$ to get a linear system $L(\mathbf{o}) = \mathbf{h}$.
  if $L(\mathbf{o}) = h$ has solutions **then**
    Calculate an assignment $\mathbf{o}$ to the oil variables such that $L(\mathbf{o}) = \mathbf{h}$
    Assign the entries of $\mathbf{o}$ to the last m entries of $\mathbf{s}'$
  end if
end while
$\mathbf{s} \leftarrow \mathcal{T}^{-1}(\mathbf{s}')$
return $\mathbf{s}$

Algorithm 2. The UOV signature generation algorithm

Algorithm UOVVerify

input: $\mathcal{P}$ — A public key
M — A message
$\mathbf{s}$ — A candidate–signature
output: **True** if $\mathbf{s}$ is a valid signature for M, **False** otherwise

$\mathbf{h} \leftarrow \mathcal{H}(M)$
$\mathbf{h}' \leftarrow \mathcal{P}(\mathbf{s})$
if $\mathbf{h} = \mathbf{h}'$ **then**
  return True
else
  return False
end if

Algorithm 3. The UOV signature verification algorithm

Thomae and Wolf [27] to reduce this to finding a solution of a polynomial system with $m + 1 - \lfloor n/m \rfloor$ equations. The best known algorithms to solve this problem use the hybrid approach [5] which was briefly described in Sect. 2. Empirically, the systems that have to be solved behave like semi-regular systems [12], therefore we can calculate the degree of regularity and use this to estimate the complexity of the hybrid approach. Petzoldt [22] uses a similar method to estimate the complexity of a direct attack against UOV, the only difference being that we have used an updated estimate of the complexity of F_5 [6]. In Petzoldt's thesis it was shown that the estimated complexity of a direct attack agrees very well with the measured complexity of a direct attack against small instances of UOV. These experiments justify ignoring the big-O notation in formula (1) and treating the formula as an estimate for the concrete hardness of the hybrid approach.

Example 1. We will estimate the complexity of a direct attack against UOV with the parameter set $(q = 31, m = 52, v = 104)$; this set is proposed in [22] as a set that achieves 128-bit security. Using the trick of Thomae et al. we can reduce finding a solution to this underdetermined system to finding a solution of a determined system with $52 + 1 - \lfloor (52 + 104)/52 \rfloor = 50$ equations. We assume this system to be semi-regular. If we fix k extra variables the degree of regularity is equal to the degree of the first term in the power series of

$$S_{50,50-k}(x) = \frac{(1-x^2)^{50}}{(1-x)^{50-k}}$$

which has a non-positive coefficient. For $k = 0$ we have $S_{50,50}(x) = (1+x)^{50}$, so the degree of regularity is 51. For $k = 1$ we have

$$S_{50,49}(x) = 1 + 49x + 1175x^3 + \cdots + 4861946401452x^{25} - 4861946401452x^{26} + O(x^{27}),$$

where all the omitted terms have positive coefficients, so the degree of regularity is 26. We can now use (1) to estimate the complexity of the hybrid approach. We prefer to err on the side of caution, so we have chosen $\omega = 2$ for the value of the linear algebra constant. For k equal to 0 and 1 this is equal to

$$\binom{50+51}{51}^2 \approx 2^{194.7} \qquad \text{and} \qquad 31\binom{50-1+26}{26}^2 \approx 2^{137.8}$$

respectively. Continuing this for higher values of k we eventually see that the optimal value of k is 6, the corresponding degree of regularity is 16 and the complexity of the direct attack is $2^{123.9}$.

In the example we concluded that the complexity of the attack is less than 2^{128} which was supposed to be the security level of the parameter set $(q = 31, m = 52, v = 104)$ according to [22]. Even though we have used roughly the same method of estimating the complexity as the method used by Petzoldt [22] we arrive at a slightly different value because we have used a tighter bound on the complexity of F_5 coming from an improved analysis of the hybrid approach [6].

With this method we can calculate the minimal number of equations that is needed in a determined semi-regular system in order to guarantee that the complexity of finding a solution is larger than a targeted security level. For quantum attackers, we can follow the same method with (2) instead of (1) for estimating the complexity of the hybrid approach. The result of these calculations for the security levels of 2^{128} and 2^{256} for different finite fields of size up to $q = 2^{100}$ are plotted in Fig. 1.

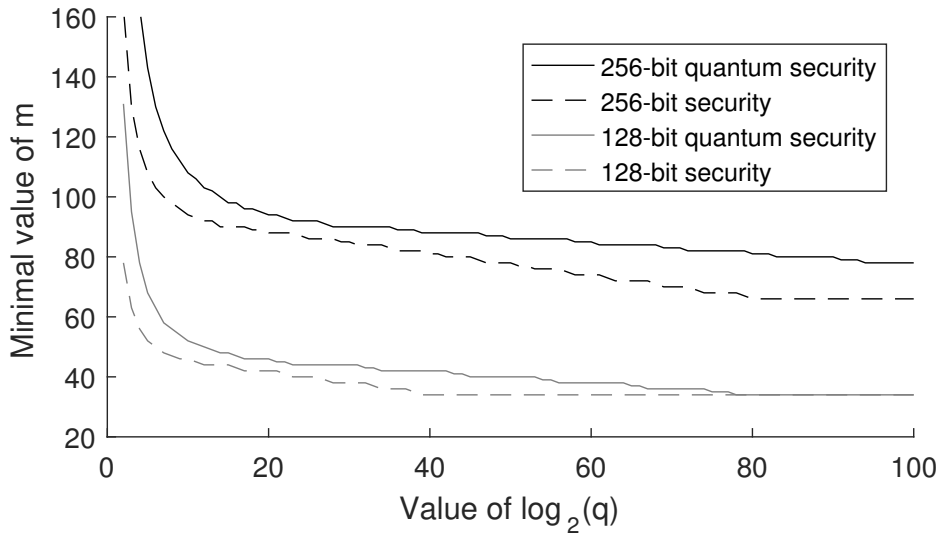

Fig. 1. The minimal sizes of determined semi-regular systems to reach 128-bit security and 256-bit security for different finite fields.

UOV attack. Patarin [20] suggested in the original version of the Oil and Vinegar scheme to choose the same number of vinegar and oil variables, or $v = m$. This choice was cryptanalyzed by Kipnis and Shamir [16]: they showed that an attacker can find the inverse image of the oil variables under the map $\mathcal{T}$. This is enough information to find an equivalent secret key, so this breaks the scheme. This approach generalizes for the case $v > m$; the complexity then increases to $O(q^{v-m}n^4)$ [15] and is thus exponential in $v - m$. Typically one chooses $v = 2m$ or $v = 3m$ to preclude the UOV attack.

UOV reconciliation attack. Similar to the UOV attack, the UOV reconciliation attack proposed by Ding et al. [9] tries to find an equivalent secret key. We present a brief summary. In this section we will make a distinction between m, the number of polynomials in the public and private system, and o, the number

of oil variables. In the UOV signature scheme these numbers are the same, which explains why we did not need to make this distinction before. It turns out that for a public key $\mathcal{P}$ there exists with a very high probability a private key $(\mathcal{F}, \mathcal{T})$ such that the matrix representation of $\mathcal{T}$ is of the form

$$\mathbf{M}_{\mathcal{T}} = \begin{pmatrix} \mathbf{I}_v & \mathbf{T} \\ 0 & \mathbf{I}_o \end{pmatrix}.$$

This means that an attacker only has to find the $v \times o$ matrix $\mathbf{T}$ to get an equivalent key. The UOV reconciliation attack tries to find $\mathbf{T}$ algebraically by solving a quadratic system. If the choice of $\mathcal{T}$ is correct (i.e. there exists a private key of the form $(\mathcal{F}, \mathcal{T})$, then we have that the matrix representation $\mathbf{P}_i$ of the quadratic part of each polynomial in the public key satisfies for all $1 \leq i \leq m$

$$\begin{pmatrix} *_{v\times v} & *_{o\times v} \\ *_{v\times o} & 0_{o\times o} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_v & 0 \\ -\mathbf{T} & \mathbf{I}_o \end{pmatrix} \mathbf{P}_i \begin{pmatrix} \mathbf{I}_v & -\mathbf{T} \\ 0 & \mathbf{I}_o \end{pmatrix}. \tag{3}$$

The condition that the lower right $o \times o$ submatrices of the private system consist of zeroes give quadratic equations in the entries of $\mathbf{T}$. It looks like we have o^2 equations for each component, but since the matrix representations are only defined up to the addition of a skew symmetric matrix this gives only $o(o+1)/2$ equations per component. In total we have a system of $mo(o+1)/2$ equations in vo variables. The reconciliation attack tries to recover $\mathbf{T}$ by solving this system of equations.

The reconciliation system has a structure that makes it much easier to solve compared to a random system of the same size. In fact, Ding et al. argue that the complexity of this attack for UOV variants with $v \leq m$ (like Rainbow and TTS) is the same as the complexity of solving a system of m equations in v variables [9].

In the case $v \geq m$ the complexity of the attack is more difficult to estimate, but we can formulate a lower bound to the complexity of the attack. The reconciliation system has $mo(o+1)/2$ equations in ov variables. For all parameter choices of UOV this is a heavily overdetermined system, so it should not be a surprise that there is only one matrix $\mathbf{T}$ that satisfies (3). Computer experiments have shown that there is a unique solution for $\mathbf{T}$ as soon as the number of equations of the reconciliation system exceeds the number of variables. Let $\mathrm{Rec}[v, o, m]$ denote the complexity of a key reconciliation attack against a UOV public system with v vinegar variables, o oil variables and m polynomials in the public key. Increasing m only makes the reconciliation attack easier. Indeed, increasing the number of equations can only make the attack easier, because an attacker could just ignore the extra equations and still find the same unique solution. In other words, if $m < m'$, then we have $\mathrm{Rec}[v, o, m] \geq \mathrm{Rec}[v, o, m']$, provided that $mo(o+1)/2 > ov$, which is the case for all good UOV parameter choices.

We can now derive a lower bound on the complexity of a reconciliation attack when $v > m = o$. According to the above observation, we can increase m, the number of equations, until it matches the number of vinegar variables v, and

this would make solving the system easier, i.e. we have

$$\mathrm{Rec}[v, m, m] \geq \mathrm{Rec}[v, m, v]\,. \tag{4}$$

We can now use the argument of Ding et al. which says that when $m \geq v$, the complexity of the reconciliation attack is equal to the complexity of solving a system of m quadratic equations in v variables, so $\mathrm{Rec}[v, m, v]$ is equal to the complexity of solving a system of v quadratic equations in v variables.

We conclude that a UOV reconciliation attack on a UOV system with m equations and $v \geq m$ vinegar variables is at least as difficult as solving a system of v quadratic variables in v equations, but it is expected to be more difficult, because a lot of hardness is lost in the inequality (4). In particular, the reconciliation attack is less effective against the UOV scheme than attacking the system $\mathcal{P}(s) = \mathcal{H}(M)$ directly.

Quantum attacks. There are no known specialized quantum algorithms that solve multivariate quadratic equations. However, as described in Sect. 2, Grover's algorithms can be used to speed up the exhaustive search part of hybrid solution finding algorithms. This quantum version of the hybrid approach algorithm can be used to speed up a direct attack and a reconciliation attack.

Grover search could be used to speed up the UOV attack from $O(q^{v-m}n^4)$ to $O(q^{\frac{v-m}{2}}n^4)$. This requires repeatedly running an algorithm that calculates the common eigenspaces of a set of matrices and checks whether any of these eigenspaces lies within the oil subspace in superposition. In comparison with the classical algorithm this has the disadvantage that it cannot be parallelized without a significant amount of overhead.

4 Improving UOV

In [22] Petzoldt presented a new method to reduce the public key size of UOV by roughly a factor 8. The key generation algorithm was adapted to make it possible to choose a large part of the public key. One can generate this part with a pseudo-random number generator and replace a large part of the public key by a seed. Also, it is possible to choose part of the public key in such a way such that signatures can be verified faster [21].

Usually, during key generation, a UOV system $\mathcal{F}$ and an invertible linear map $\mathcal{T}$ are chosen randomly, and then $\mathcal{P}$ is determined as $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$. With this strategy we have full control over $\mathcal{F}$, but no control over the public key $\mathcal{P}$. Instead, Petzoldt proposed to first pick $\mathcal{T}$ and $v(v+1)/2+mv$ coefficients of each polynomial of $\mathcal{P}$. Then we solve the system $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ to find the coefficients of $\mathcal{F}$, and the remaining coefficients of $\mathcal{P}$. This is a linear system of equations, so this can happen efficiently. With a small probability this system does not have any solutions, but in that case we can simply try again with a different choice of $\mathcal{T}$. For the details of this method we refer to [22].

With this approach the public key size is decreased with $m(v(v+1)/2+mv)$ field elements, at the negligible cost of including the seed for the random number generator. The public key size is now $m^2(m+1)/2\log_2(q) + |\text{seed}|$. Table 1 shows that this method drastically reduces the size of the public key. However, the public key remains much larger than the signature schemes that are in use today such as RSA [25] and DSA [17], which typically stay well under 1 kB. Note that if Petzoldt's method is used, the size of the public key is independent of v, the number of vinegar variables.

Table 1. The effect of Petzoldt's method on the public key size

security level	q	(m, v)	public key (kB)	public key with Petzoldt's method (kB)
100-bit	2^8	(36,72)	207	23
128-bit	2^8	(47,94)	460	52
192-bit	2^8	(72,144)	1648	185
256-bit	2^8	(98,196)	4150	464

5 Lifting $\mathcal{P}$ to an extension field

In this section we will work with UOV over a finite field $\mathbb{F}_{2^r}$ of characteristic 2. The parameter r is quite important for the security of the scheme, the signature size and key sizes. It can be seen in Fig. 1 that by choosing a larger value of r we can put a smaller number of equations in the system and still reach the same level of security. Since the number of field elements in the public key and secret key is $O(m^3)$ it is desirable to have a small value of m. However, since it costs r bits to store a field element r should not be too big either. We must make a trade-off between large r and large m. In this section we propose a scheme that gets some security benefits of a high value of r, but has a public and private key with coefficients in $\mathbb{F}_2$, greatly reducing the key sizes.

5.1 Description of the new scheme

As usual, the public key of the scheme represents a quadratic system over $\mathbb{F}_{2^r}$, given by

$$\mathcal{P} = \mathcal{F} \circ \mathcal{T}\,.$$

When we want to sign a message m we use a hash function to generate a digest of mr bits which represents a vector $\mathbf{h}$ of m elements of $\mathbb{F}_{2^r}$. Then we use the knowledge of the private key to solve the system $\mathcal{P}(\mathbf{s}) = \mathbf{h}$ to get a valid signature $\mathbf{s}$. However, the difference with standard UOV is that we now choose all the coefficients of $\mathcal{F}, \mathcal{P}$ and $\mathcal{T}$ in $\mathbb{F}_2$. Therefore the key generation process is identical to the key generation process of a regular UOV scheme over $\mathbb{F}_2$. In particular, we can use the approach of Petzoldt [22] as explained in Sect. 4 to

reduce the size of the public key. Contrary to the key generation, the signature generation and verification still happen over the field $\mathbb{F}_{2^r}$ as usual.

To summarize, we simply take a key pair of the UOV scheme over $\mathbb{F}_2$, and use it as a key pair for the UOV scheme over $\mathbb{F}_{2^r}$. The public key is thus approximately a factor r smaller than if we were to use the regular UOV scheme over $\mathbb{F}_{2^r}$ since we only use one bit to represent each coefficient instead of r bits. Furthermore, we can now choose r to be much larger than what would otherwise its optimal value. This in turn allows for a smaller value of m (See Fig. 1), reducing the public key size even more.

The public key consists of a seed for a pseudorandom number generator and the part of the public map which cannot be generated. The total size of the public key is therefore

$$|\text{seed}| + \frac{m^2(m+1)}{2} \text{ bits.}$$

Storing the private maps $\mathcal{F}$ and $\mathcal{T}$ would take

$$m\frac{v(v+1)}{2} + m^2 v \text{ bits and } n^2 \text{ bits}$$

respectively, but they do not need to be stored, because they can be calculated using the key generation algorithm each time they are needed. A signature consists of $n = m + v$ elements of $\mathbb{F}_{2^r}$, so the size of the signature is nr bits.

Remark 2. Though we have presented this scheme with a finite field of characteristic 2 and with the subfield $\mathbb{F}_2 \subset \mathbb{F}_{2^r}$, it is easy to see that we can use this scheme with any field extension of finite fields $K \subset K'$. In such a scenario we generate a key pair with coefficients in the small field K, and the signing and verifying is done with elements of the big field K'.

5.2 Security analysis of the new scheme

Direct attack. This attack tries to forge a signature for a certain message M by trying to find a solution $s \in \mathbb{F}_{2^r}^n$ for the system $\mathcal{F}(s) = \mathcal{H}(M)$. The best known methods for this use the hybrid approach as described in Sect. 2.

For a direct attack against the new scheme all the coefficients of the system that needs to be solved lie in $\mathbb{F}_2$, except those of the constant terms, because those coefficients come from the message digest. We claim that this does not significantly reduce the hardness of finding solutions relative to the case where the coefficients are generic elements of $\mathbb{F}_{2^r}$. It has been noticed by Faugère and Perret [12] that the polynomial systems that result from fixing $\approx v$ variables in a UOV system behave like semi-regular systems. The degree of regularity of a quadratic semi-regular system is given by the degree of the first term in the power series of

$$\frac{(1-x^2)^m}{(1-x)^n}$$

with a non-positive coefficient. In particular the degree of regularity does not depend on q for semi-regular systems. Hence, the degree of regularity for a direct

attack against the modified UOV scheme is identical to the degree of regularity of an attack against the regular UOV scheme. Therefore a Gröbner basis computation against the modified scheme is not significantly more efficient than a Gröbner basis computation against regular UOV with the same parameters. This argument is confirmed by the experimental data in Table 2. There we see that a direct attack is slightly faster against the modified scheme than against the original UOV scheme, but only by a small constant factor. Even though the Gröbner basis is computed over $\mathbb{F}_{2^r}$, the largest part of the arithmetic only involves the field elements 0 and 1, so the arithmetic is faster than with generic elements of $\mathbb{F}_{2^r}$. This is where the difference observed in Table 2 comes from. If we do the same experiment with a smaller extension field such as $\mathbb{F}_{2^8}$ there is no observed difference between the running time of a direct attack against a regular UOV scheme and our modified scheme.

Remark 3. In a direct attack one fixes $\approx v$ variables randomly to make the system a slightly overdetermined system. In our experiments we have fixed these variables to values in $\mathbb{F}_2$ to make sure that we do not introduce linear terms with coefficients in $\mathbb{F}_{2^r}$ instead of $\mathbb{F}_2$ in the case of the modified UOV scheme.

Table 2. Running time of a direct attack against the regular UOV scheme over $\mathbb{F}_{2^{64}}$ and the modified UOV scheme, with the MAGMA v2.22-10 implementation of the F4 algorithm. We did not implement the method of Thomae and Wolf [27].

(m,v)	Regular UOV (s)	Lifted UOV (s)	difference
(7,35)	0.43	0.21	-52%
(8,40)	1.56	0.76	-51%
(9,45)	7.00	3.21	-54%
(10,50)	33.50	17.44	-48%
(11,55)	132.88	76.60	-42%
(12,60)	828.31	588.33	-29%

Remark 4. It might seem tempting to decompose the equations over $\mathbb{F}_{2^r}$ into equations over $\mathbb{F}_2$ to make a direct attack more efficient. This decomposition is done by fixing some basis $\beta_1, \cdots, \beta_r$ of $\mathbb{F}_{2^r}$ over $\mathbb{F}_2$ and replacing each variable x_i by $\sum_{j=1}^{r} \hat{x}_{i,j}\beta_j$, where the $\hat{x}_{i,j}$ are nr new variables in $\mathbb{F}_2$. Each equation of the original system is then decomposed into r equations, resulting in a total of mr equations in nm variables over $\mathbb{F}_2$. The problem with this approach is that the number of equations and variables is increased by the factor r, which makes the naive approach of solving the decomposed system with a generic boolean solver hopelessly slow. However, the decomposed system has a specific structure which could potentially be exploited to solve the system more efficiently. We investigated this possibility, but we we were not able to make any progress. It should be pointed out that this idea does not only apply to our scheme, but to any multivariate cryptosystem over a field of non-prime order. Still, no such attacks are reported in literature. One could say that the idea of decomposing

a system to make it easier to solve is not very promising because in big-field schemes such as Gui [24] and medium-field schemes such as HMFEv [23] the systems are decomposed with the objective of making them *harder* to solve for an attacker.

Key recovery attacks. In contrast to a direct attack, the modified scheme is more vulnerable to a key recovery attack. Since the key pair used in the Lifted UOV scheme is identical to the key pair of regular UOV over the field $\mathbb{F}_2$ it is clear that a key recovery attack against the Lifted UOV scheme is equivalent to a key recovery attack against a regular UOV scheme over $\mathbb{F}_2$, which is much easier than a key recovery attack against UOV over $\mathbb{F}_{2^r}$. Luckily, key recovery attacks against UOV have been investigated ever since the invention of the oil and vinegar scheme in 1997 [20], so it is well understood which attacks are possible (see Sect. 3.2) and what the complexities of these attacks are. It is also clear that we can make key recovery attacks harder by increasing the number of vinegar variables.

The UOV attack attempts to recover an equivalent private key by searching for the oil subspace. This attack has complexity $q^{v-m-1} \cdot n^4$. Since a UOV attack on the Lifted UOV scheme is equivalent to a UOV attack over $\mathbb{F}_2$, we have that the complexity of a UOV attack against the Lifted UOV scheme is $2^{v-m-1} \cdot n^4$.

The reconciliation attack against the lifted UOV scheme is equivalent to the UOV reconciliation attack against UOV over the field $\mathbb{F}_2$. A lower bound on the complexity of this attack is given by the complexity of solving a quadratic system of v variables and v equations over $\mathbb{F}_2$, but we expect the problem to be harder. There exists specialized algorithms for solving polynomial systems over $\mathbb{F}_2$ that are more efficient than the generic hybrid approach. One method is a smart exhaustive search, which requires approximately $log_2(n)2^{n+2}$ bit operations [7]. The BooleanSolve algorithm [3] combines an exhaustive search with sparse linear algebra to achieve a complexity of $O(2^{0.792n})$. However the method only becomes faster than the exhaustive search method when $n > 200$. Recently, Joux et al. proposed a new algorithm that was able to solve a boolean system of 146 quadratic equations in 73 variables in one day [14]. The algorithm beats the exhaustive search algorithm, even for small systems. The complexity of this algorithm is still under investigation, but a rough estimate based on the reported experiments suggests that it scales like $2^{\alpha n}$ with α between 0.8 and 0.85 and with a small constant factor. For choosing the parameters of our signature scheme, we have assumed that a determined system of n quadratic boolean equations provides $0.75n$ bits of security, even though this is likely to seriously overestimate the capabilities of the state of the art algorithms. Quantum attackers can use Grover search to solve systems over $\mathbb{F}_2$ with complexity $O(2^{n/2})$.

5.3 Choice of parameters

For convenience and efficiency we will work with binary finite fields whose elements are represented by a number of bits that is a multiple of 16, i.e. the finite fields we want to use are $\mathbb{F}_{2^{16}}, \mathbb{F}_{2^{32}}, \mathbb{F}_{2^{48}}$ and so on.

When designing a signature scheme of security level l, we choose a finite field that is large enough such that the minimal number of equations in a determined regular system that is needed to reach the security level l is minimized. Figure 1 shows that for 128-bit and 256-bit security the chosen fields are $\mathbb{F}_{2^{48}}$ and $\mathbb{F}_{2^{80}}$ respectively, and the minimal number of equations is 34 and 66 respectively or 40 and 81 when considering quantum attacks. For 100-bit and 192-bit security the chosen fields are $\mathbb{F}_{2^{32}}$ and $\mathbb{F}_{2^{64}}$, and the minimal number of equations is 27 and 50 for classical attackers or 33 and 60 for quantum attackers.

We now consider the constraints on the parameters due to the different attacks against our scheme. In order to be safe against a direct attack we require

$$m - \lfloor v/m \rfloor \geq m_{min}\,,$$

with m_{min} equal to $27, 34, 50$ or 66 if the desired security level is 100 bits, 128 bits, 192 bits, or 256 bits respectively. For quantum attackers m_{min} is equal to $33, 40, 60$ and 81 respectively. In order to be safe against the UOV attack we require

$$2^{v-m-1}n^4 > 2^l \qquad \text{or} \qquad 2^{(v-m-1)/2}n^4 > 2^l\,,$$

depending on whether we want l bits of security against classical, or quantum adversaries. To be secure against the UOV reconciliation attack it suffices that an attacker cannot solve a determined system with v equations over $\mathbb{F}_2$. Therefore it suffices to have

$$2^{0.75v} > 2^l \qquad \text{or} \qquad 2^{v/2} > 2^l$$

for classical and quantum attackers respectively. The parameter sets displayed in Table 3 satisfy all the constraints for the targeted security level and minimize the size of the public key, i.e. they minimize m. In the last column of the table, the bit complexity of the best known classical attack against the parameter set is calculated. For all the proposed parameters the best known classical attack is a direct Groebner basis attack.

5.4 Trade-off

In comparison to regular UOV, Lifted UOV has much smaller public keys, but also larger signatures. In the discussion above, we have chosen the parameter r very large in order to minimize the size of the public key, without considering the size of the signatures. It is possible to make a trade-off between the size of the public key and the size of the signature by choosing a smaller value of r. Having a smaller value of r requires a larger value of m to reach the same security level, resulting in a larger public key, but since the signature consists of n elements of $\mathbb{F}_{2^r}$ it also leads to smaller signatures. Figure 2 compares public key sizes and

Table 3. Parameter choices and corresponding public key and signature sizes for different security levels

security level		(r, m, v)	|pk| (kB)	|sig| (kB)	classical security
100-bit	classical	(32,31,134)	1.9	0.6	
	quantum	(32,37,200)	3.2	0.9	115 bit
128-bit	classical	(48,38,171)	3.4	1.2	
	quantum	(48,45,256)	5.7	1.8	153 bit
192-bit	classical	(64,54,256)	9.8	2.4	
	quantum	(64,65,384)	17.0	3.5	224 bit
256-bit	classical	(80,70,341)	21.2	4.0	
	quantum	(80,87,526)	40.7	6.0	296 bit

signature sizes of the Lifted UOV scheme with different values of the parameter r with some other MQ signature schemes [22], the lattice-based signature scheme BLISS-II [10] and SPHINCS, a hash-based signature scheme [4]. Note that even though the MQ schemes UOVRand and RainbowLRS2 claim to provide 128-bit of post-quantum security, their parameters are not chosen to resist quantum attacks on the MQ problem or quantum versions of the UOV attack. So we are not comparing schemes with the same security level. Ignoring quantum attacks, the Lifted UOV signature scheme with $r = 48$ in the comparison achieves 153 bits of security.

Example 2. For some application on a low-cost device it might be desirable to have a signature scheme that provides 128 bits of post-quantum security with minimal signature sizes subject to the condition that the public key is smaller than, say, 10 kB. If we choose the parameters as in the discussion above, we would have a public key of 5.7 kB and signatures of 1.8 kB. However, we can do better by choosing $r = 12$. The lowest values of m and v providing 128 bits of security are then $m = 54$ and $v = 256$. This leads to a public key of 9.8 kB (< 10kB) and a signature of 0.45 kB.

6 Implementation and results

We developed an ANSI C implementation of the Lifted UOV signature scheme. The large fields are implemented as extension fields of $\mathbb{F}_{2^{16}}$ and the arithmetic in $\mathbb{F}_{2^{16}}$ is done using log tables. We have a table that maps each nonzero element x to the number y such that $x = a^y$, where a is some generator of the group $\mathbb{F}_{2^{16}}^{\times}$. Conversely, we also have a table that maps a number y to the element a^y. Multiplication in $\mathbb{F}_{2^{16}}$ is then computed with three table lookups and an addition modulo $2^{16} - 1$. Note that this approach could make our implementation vulnerable to cache timing attacks. Newer CPUs support the CLMUL instruction set which could be used to perform the field arithmetic efficiently without the need for lookup tables, eliminating the possibility of this attack. Two field elements are added using a XOR operation. During the key generation phase we only use

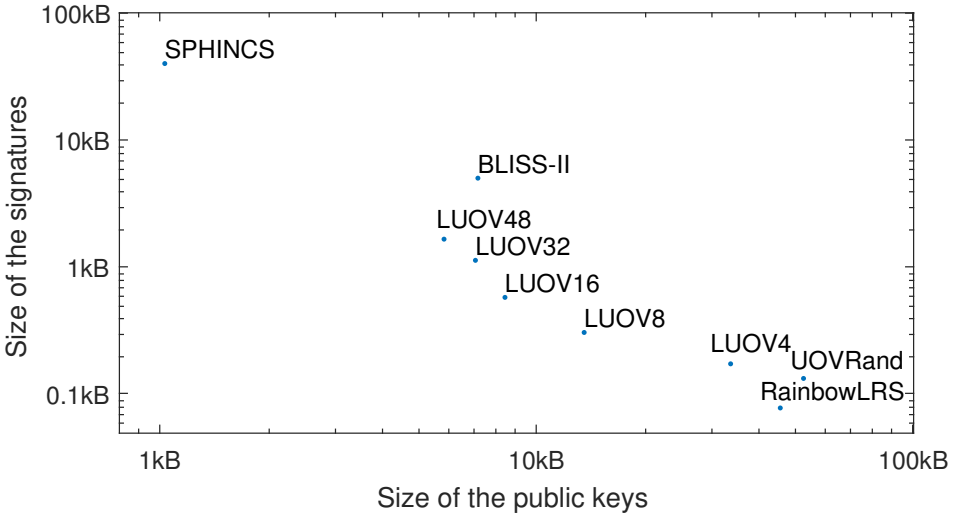

Fig. 2. Comparison of different signature schemes providing 128 bits of post-quantum security.

elements of $\mathbb{F}_2$, so we have used bit slicing whenever possible to speed up the algorithm. The running times of the key generation, signature generation and the verification algorithms are displayed in Table 4.

Please note that the implementation uses naive implementations of matrix multiplication, polynomial multiplication and Gaussian reduction, and the code was not heavily optimized. Therefore, it can be expected that the running times reported in Table 4 are nowhere near optimal. Some techniques that can speed up the code very significantly include writing cache friendly code, using parallelization and using Karatsuba's algorithms for the field arithmetic. Moreover, it is possible to use a method of Petzoldt to structure part of the public key in such a way that the verification algorithm is faster [22]. In order to avoid storing the large private key, part of the key generation algorithm is run each time a signature is generated to generate the private key. If a batch of messages is signed together this step only has to happen once. Alternatively, if storing the private key is not an issue, this part can be omitted altogether to speed up the signing algorithm significantly.

7 Conclusion

The simple idea of lifting a UOV key pair from $\mathbb{F}_2$ to an extension field $\mathbb{F}_{2^r}$ increases the security against direct attacks without affecting the size of the public key. At the same time, thanks to the method of Petzoldt, we can increase the number of vinegar variables to protect against key recovery attacks without

Table 4. Running times for the key generation, signing and verification algorithms on a single thread on an Intel®Core™ i7-4710MQ CPU at 2.5 GHz

security level		key gen (ms)	sig gen (ms)	verification (ms)
100-bit	classical	4	6	3
	quantum	13	16	7
128-bit	classical	10	14	7
	quantum	26	34	15
192-bit	classical	32	46	21
	quantum	148	156	54
256-bit	classical	125	149	55
	quantum	366	410	144

increasing the size of the public key. These two ideas come together to create a secure signature scheme whose public key is an order of magnitude smaller than other MQ signature schemes, with slightly larger signatures. The signature scheme is very competitive with other post-quantum signature schemes. By choosing the parameter r it is possible to make a trade-off between larger public keys and smaller signatures or vice versa. We developed a rudimentary ANSI C implementation of the Lifted UOV signature scheme which shows that key generation, signing and verification takes only a few milliseconds for 100-bit security instantiations of the scheme and up to a few hundred milliseconds for 256-bit security instantiations. However it is very likely that these times can be improved significantly with an optimized implementation.

The idea of lifting keys to a large extension field can be applied to any MQ signature scheme, but it might not always be useful to produce smaller public keys. We believe that the idea could be used to improve the Rainbow signature scheme, but not HFE or C*. This is because the public keys of signature schemes such as HFE and C* are not semi-regular maps [11] and have a much smaller degree of regularity than random maps of the same dimensions. This means that guessing a few variables does not necessarily reduce the degree of regularity, like it does in the case of semi-regular systems. This makes the hybrid approach unsuitable for attacking these systems, since solving the system with one big Gröbner basis computation is more efficient. Therefore there is no point in lifting the system to a larger field, because the complexity of a Gröbner basis computation is largely independent of the size of the finite field.

Acknowledgements This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the European Commission through the ICT programme under contract FP7-ICT-2013-10-SEP-210076296 PRACTICE, through the Horizon 2020research and innovation programme under grant agreement No H2020-ICT-2014-644371 WITDOM and H2020-ICT-2014-645622 PQCRYPTO. Ward Beullens is funded by a research grant of the KU Leuven.

References

1. PQCRYPTO ICT-645622 (2015), http://pqcrypto.eu.org/
2. Bardet, M.: Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie. Ph.D. thesis, Université Pierre et Marie Curie-Paris VI (2004)
3. Bardet, M., Faugère, J.C., Salvy, B., Spaenlehauer, P.J.: On the complexity of solving quadratic Boolean systems. Journal of Complexity 29(1), 53–75 (2013)
4. Bernstein, D.J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Schneider, M., Schwabe, P., Wilcox-O'Hearn, Z.: SPHINCS: practical stateless hash-based signatures. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 368–397. Springer (2015)
5. Bettale, L., Faugere, J.C., Perret, L.: Hybrid approach for solving multivariate systems over finite fields. Journal of Mathematical Cryptology 3(3), 177–197 (2009)
6. Bettale, L., Faugère, J.C., Perret, L.: Solving polynomial systems over finite fields: Improved analysis of the hybrid approach. In: Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation. pp. 67–74. ACM (2012)
7. Bouillaguet, C., Chen, H.C., Cheng, C.M., Chou, T., Niederhagen, R., Shamir, A., Yang, B.Y.: Fast exhaustive search for polynomial systems in $\mathbb{F}_2$. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 203–218. Springer (2010)
8. Diem, C.: The XL-algorithm and a conjecture from commutative algebra. In: Asiacrypt. vol. 4, pp. 338–353. Springer (2004)
9. Ding, J., Yang, B.Y., Chen, C.H.O., Chen, M.S., Cheng, C.M.: New differential-algebraic attacks and reparametrization of Rainbow. In: International Conference on Applied Cryptography and Network Security. pp. 242–257. Springer (2008)
10. Ducas, L., Durmus, A., Lepoint, T., Lyubashevsky, V.: Lattice signatures and bimodal gaussians. In: Advances in Cryptology–CRYPTO 2013, pp. 40–56. Springer (2013)
11. Faugere, J.C., Joux, A.: Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In: Annual International Cryptology Conference. pp. 44–60. Springer (2003)
12. Faugère, J.C., Perret, L.: On the security of UOV. IACR Cryptology ePrint Archive 2009, 483 (2009)
13. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219. ACM (1996)
14. Joux, A., Vitse, V.: A crossbred algorithm for solving Boolean polynomial systems. IACR Cryptology ePrint Archive 2017, 372 (2017)
15. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 206–222. Springer (1999)
16. Kipnis, A., Shamir, A.: Cryptanalysis of the oil and vinegar signature scheme. In: Annual International Cryptology Conference. pp. 257–266. Springer (1998)
17. Kravitz, D.W.: Digital signature algorithm (Jul 27 1993), US Patent 5,231,668
18. Michael, R.G., David, S.J.: Computers and intractability: a guide to the theory of NP-completeness. WH Free. Co., San Fr (1979)
19. National Institute for Standards and Technology (NIST): Post-quantum crypto standardization (2016), http://csrc.nist.gov/groups/ST/post-quantum-crypto/

20. Patarin, J.: The oil and vinegar signature scheme. In: Dagstuhl Workshop on Cryptography1997 (1997)
21. Petzoldt, A.: Hybrid approach for the fast verification for improved versions of the UOV and Rainbow signature schemes. IACR Cryptology ePrint Archive 2013, 315 (2013)
22. Petzoldt, A.: Selecting and Reducing Key Sizes for Multivariate Cryptography. Ph.D. thesis, TU Darmstadt (Jul 2013), referenten: Professor Dr. Johannes Buchmann, Professor Jintai Ding, Ph.D.
23. Petzoldt, A., Chen, M.S., Ding, J., Yang, B.Y.: Hmfev-an efficient multivariate signature scheme. In: International Workshop on Post-Quantum Cryptography. pp. 205–223. Springer (2017)
24. Petzoldt, A., Chen, M.S., Yang, B.Y., Tao, C., Ding, J.: Design principles for hfev-based multivariate signature schemes. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 311–334. Springer (2015)
25. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21(2), 120–126 (1978)
26. Shor, P.W.: Polynomial time algorithms for discrete logarithms and factoring on a quantum computer. In: International Algorithmic Number Theory Symposium. vol. 877, pp. 289–289. Springer (1994)
27. Thomae, E., Wolf, C.: Solving underdetermined systems of multivariate quadratic equations revisited. In: International Workshop on Public Key Cryptography. pp. 156–171. Springer (2012)
28. Zalka, C.: Grover's quantum searching algorithm is optimal. Physical Review A 60(4), 2746 (1999)