Rainbow, a New Multivariable Polynomial Signature Scheme

Jintai Ding[1] and Dieter Schmidt[2]

[1] Department of Mathematical Sciences
University of Cincinnati
Cincinnati, OH, 45221, USA
ding@math.uc.edu

[2] Department of Electrical & Computer Engineering and Computer Science
University of Cincinnati
Cincinnati, OH, 45221, USA
dieter.schmidt@uc.edu

Abstract. Balanced Oil and Vinegar signature schemes and the unbalanced Oil and Vinegar signature schemes are public key signature schemes based on multivariable polynomials. In this paper, we suggest a new signature scheme, which is a generalization of the Oil-Vinegar construction to improve the efficiency of the unbalanced Oil and Vinegar signature scheme. The basic idea can be described as a construction of multi-layer Oil-Vinegar construction and its generalization. We call our system a Rainbow signature scheme. We propose and implement a practical scheme, which works better than Sflash^{v_2}, in particular, in terms of signature generating time.

Keywords: public-key, multivariable, quadratic polynomials, Oil and Vinegar

1 Introduction

The subject we deal with here are generalizations of the Oil-Vinegar construction of public key authentication systems. It is part of a general effort to build secure and efficient public key authentication systems for practical applications, in particular, low cost smart cards. The key point of our work is the idea of a multi-layer Oil-Vinegar system. The main achievement is the creation of a multi-layer Oil-Vinegar system, which we call Rainbow. We show, that the system should be more secure and more efficient than any comparable system. The importance of the work lies in the potential application of the Rainbow system as a strongly secure and very efficient public key authentication system.

Since the arrival of the RSA cryptosystem people have been trying to build new public key cryptosystems. This includes systems based on multivariable polynomials. In particular, cryptosystems based on quadratic polynomials have undergone an intensive development in the last 10 years. The theoretical basis for these constructions is the proven theorem that solving a set of multivariable polynomial equations over a finite field, in general, is an NP-hard problem, although it does not necessarily guarantee the security of a multivariable cryptosystem.

J. Ioannidis, A. Keromytis, and M.Yung (Eds.): ACNS 2005, LNCS 3531, pp. 164–175, 2005.

This direction of research attracted a lot of attention with the appearance of the construction by Matsumoto and Imai [MI88]. However, Patarin [Pat95] proved that this scheme is insecure under an algebraic attack using linearization equations. Since then Patarin and his collaborators have made a great effort to develop secure multivariable cryptosystems.

One particular direction, which Patarin and his collaborators have pursued, is inspired by the linearization equations themselves. This type of construction includes Little Dragon, Dragon, Oil and Vinegar, Unbalanced Oil and Vinegar [Pat96,KPG99]. The construction of the last two schemes uses the idea that certain quadratic equations can be easily solved if we are allowed to guess a few variables. Let k be a finite field. The key construction is a map F from k^{o+v} to k^o:

$$F(x_1, .., x_o, x'_1, .., x'_v) = (F_1(x_1, .., x_o, x'_1, .., x'_v), \ldots, F_o(x_1, .., x_o, x'_1, .., x'_v)),$$

and each F_l is in the form:

$$F_l(x_1, \ldots, x_o, x'_1, \ldots, x'_v) = \sum a_{l,i,j} x_i x'_j + \sum b_{l,i,j} x'_i x'_j + \sum c_{l,i} x_i + \sum d_{l,j} x'_j + e_l$$

where x_i, $i = 1, \ldots, o$, are the Oil variables and x'_j, $j = 1, \ldots, v$, are the Vinegar variables in the finite field k. (Note the similarity of the above formula with the linearization equations.) We call such a type of polynomial an 'Oil and Vinegar polynomial'. The reason that it is called Oil and Vinegar scheme is due to the fact that in the quadratic terms the Oil and Vinegar variables are not fully mixed (like oil and vinegar). This allows us to find one solution easily for any equation of the form

$$F(x_1, \ldots, x_o, x'_1, \ldots, x'_v) = (y_1, \ldots, y_o),$$

when $(y_1, \ldots, y_o)$ is given. To find one solution, one just needs to randomly choose values for the Vinegar variables and plug them into the equations above, which will produce a set of o linear equations with o variables. This should, with a probability close to 1, give us a solution. If it does not, one can try again by selecting different values for the Vinegar variables, until one succeeds in finding a solution.

This family of cryptosystems is designed specifically for signature schemes, where we need only to find one solution for a given set of equations and not a unique solution.

Once we have this map F, we "hide" it by composing it from the left and the right sides by two invertible affine linear maps L_1 and L_2, in the same way as it was done in the construction of [MI88]. Since L_1 is on k^o and L_2 on k^{o+v}, this generates a quadratic map

$$\bar{F} = L_1 \circ F \circ L_2$$

from k^{o+v} to k^o ($\circ$ means composition of two maps).

The balanced Oil and Vinegar scheme is characterized by $o = v$, but it was defeated by Kipnis and Shamir [KS99] using matrices related to the bilinear forms defined by quadratic polynomials.

For the unbalanced Oil and Vinegar scheme, $v > o$, it was shown in [KPG99] that a specific attack has a complexity of roughly $q^{v-o-1}o^4$, when $v \approx o$. This means, that if o is not too large (< 100) and a given fixed field of size q, then $v - o$ should be large enough, but also not too large, to ensure the security of the scheme.

However, one must notice that in this scheme the document to be signed is a vector in k^o and the signature is a vector in k^{o+v}. This means that the signature is at least twice the size of the document and with a large $v + o$ the system becomes less efficient.

We propose in this paper a new construction that uses the Oil and Vinegar construction multiple times such that in the end the signature will be only slightly longer than the document. This scheme is therefore much more efficient. It is called Rainbow.

In the next section, we present the general construction and a practical example. Then we give a general cryptanalysis. We compare our scheme with Sflash and the original unbalanced Oil and Vinegar schemes. Finally we discuss ways to optimize the scheme and to generalize it further.

2 Rainbow, a Signature Scheme

In this section, we present first the general construction of Rainbow and then give an example of its practical implementation.

2.1 General Construction of Rainbow

Let S be the set $\{1, 2, 3, \ldots, n\}$. Let $v_1, \ldots, v_u$ be u integers such that $0 < v_1 < v_2 < \cdots < v_u = n$, and define the sets of integers $S_l = \{1, 2, \ldots, v_l\}$ for $l = 1, \ldots, u$, so that we have

$$S_1 \subset S_2 \subset \cdots \subset S_u = S.$$

The number of elements in S_i is v_i.

Let

$$o_i = v_{i+1} - v_i, \text{ for } i = 1, \ldots, u-1.$$

Let O_i be the set such that

$$O_i = S_{i+1} - S_i, \text{ for } i = 1, \ldots, u-1.$$

Let P_l be the linear space of quadratic polynomials spanned by polynomials of the form

$$\sum_{i \in O_l, j \in S_l} \alpha_{i,j} x_i x_j + \sum_{i,j \in S_l} \beta_{i,j} x_i x_j + \sum_{i \in S_{l+1}} \gamma_i x_i + \eta$$

We can see that these are Oil and Vinegar type of polynomials such that x_i, $i \in O_l$ are the Oil variables and x_i, $i \in S_l$ are the Vinegar variables. We call x_i, $i \in O_l$ an l-th layer Oil variable and x_i, $i \in S_l$ an l-th layer Vinegar variable.

We call any polynomial in P_l an l-th layer Oil and Vinegar polynomial. Clearly we have $P_i \subset P_j$ for $i < j$.

In this way, each P_l, $l = 1, \ldots, u-1$ is a set of Oil and Vinegar polynomials. Each polynomial in P_l has x_i, $i \in O_l$ as its Oil variables and x_i, $i \in S_l$ as its Vinegar variables. The Oil and Vinegar polynomials in P_i can be defined as polynomials such that $x_i \in O_i$ are the Oil variables and x_i, $i \in S_i$ are the Vinegar variables. This can be illustrated by the fact that

$$S_{i+1} = \{S_i, O_i\}.$$

Now, we will define the map F of the Rainbow signature scheme. It is a map F from k^n to k^{n-v_1} such that

$$F(x_1, \ldots, x_n) = (\tilde{F}_1(x_1, \ldots, x_n), \ldots, \tilde{F}_{u-1}(x_1, \ldots, x_n))$$
$$= (F_1(x_1, \ldots, x_n), \ldots, F_{n-v_1}(x_1, \ldots, x_n)),$$

each $\tilde{F}_i$ consists of o_i randomly chosen quadratic polynomials from P_i. By a randomly chosen polynomial, we mean that we choose its coefficients at random.

In this way, we can see that F actually has $u-1$ layers of Oil and Vinegar constructions. The first layer consists of o_1 polynomials $F_1, \ldots, F_{o_1}$ such that x_j, $j \in O_1$ are the Oil variables and x_j, $j \in S_1$ are the Vinegar variables. The i-th layer consists of o_i polynomials, $F_{v_i+1}, \ldots, F_{v_{i+1}}$, such that x_j, $j \in O_i$ are the Oil variables and x_j, $j \in S_i$ are the Vinegar variables. From this, we can build a rainbow of our variables:

$$[x_1, \ldots, x_{v_1}]; \{x_{v_1+1}, \ldots, x_{v_2}\}$$
$$[x_1, \ldots, x_{v_1}, x_{v_1+1}, \ldots, x_{v_2}]; \{x_{v_2+1}, \ldots, x_{v_3}\}$$
$$[x_1, \ldots, x_{v_1}, x_{v_1+1}, \ldots, x_{v_2}, x_{v_2+1}, \ldots, x_{v_3}]; \{x_{v_3+1}, \ldots, x_{v_4}\}$$
$$\ldots; \ldots$$
$$[x_1, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, x_{v_{u-1}}]; \{x_{v_{u-1}+1}, \ldots, x_n\}$$

Each row above represents a layer of the Rainbow. For the l-th layer above, the ones in [ ] are Vinegar variables, the ones in { } are Oil variables and each layer's Vinegar variables consists of all the variables in the previous layer.

We call F a Rainbow polynomial map with $u-1$ layers.

Let L_1 and L_2 be two randomly chosen invertible affine linear maps, L_1 is on k^{n-v_1} and L_2 on k^n.

Let

$$\bar{F}(x_1, \ldots, x_n) = L_1 \circ F \circ L_2(x_1, \ldots, x_n),$$

which consists of $n - v_1$ quadratic polynomials with n variables.

We will now use the above to construct a public key **Rainbow** signature scheme.

1. Public Key

For a Rainbow signature scheme, the public key consists of the $n - v_1$ polynomial components of $\bar{F}$ and the field structure of k.

2. Private Key
The private key consists of the maps L_1, L_2 and F.

3. Signing a Document
To sign a document, which is an element $Y' = (y'_1, \ldots, y'_{n-v_1})$ in k^{n-v_1}, one needs to find a solution of the equation

$$L_1 \circ F \circ L_2(x_1, \ldots, x_n) = \bar{F}(x_1, \ldots, x_n) = Y'.$$

We can apply the inverse of L_1 first, then we have

$$F \circ L_2(x_1, \ldots, x_n) = L_1^{-1} Y' = \bar{Y}'.$$

Next we need to invert F. In this case, we need to solve the equation

$$F(x_1, \ldots, x_n) = \bar{Y}' = (\bar{y}'_1, \ldots, \bar{y}'_{n-v_1}).$$

We first randomly choose the values of $x_1, \ldots, x_{v_1}$ and plug them into the first layer of o_1 equations given by

$$\tilde{F}_1 = (\bar{y}'_1, \ldots, \bar{y}'_{o_1}).$$

This produces a set of o_1 linear equations with o_1 variables, $x_{o_1+1}, \ldots, x_{v_2}$, which we solve to find the values of $x_{o_1+1}, \ldots, x_{v_2}$. Then we have all the values of x_i, $i \in S_2$.

Then we plug these values into the second layer of polynomials, which will again produce o_2 number of linear equations, which then gives us the values of all x_i, $i \in S_3$. We repeat the procedure until we find a solution.

If at any time, a set of linear equations does not have a solution, we will start from the beginning again by choosing another set of values for $x_1, \ldots, x_{v_1}$. We will continue until we find a solution. We know from [Pat96], that with a very high probability we can expect to succeed if the number of layers is not too large.

Then we apply the inverse of L_2, which gives us a signature of Y', which we will denote by $X' = (x'_1, \ldots, x'_n)$.

4 Verifying the Signature
To verify a signature, one only needs to check if indeed

$$\tilde{F}(X') = Y'.$$

In order to sign a large document, one can go through the same procedure for Flash as in [PCG01] by applying a hash function first, then sign the hash value of the document.

2.2 A Practical Implementation of Rainbow

For a practical implementation we have chosen k to be a finite field of size $q = 2^8$.

Let $n = 33$ and S be the set $\{1, 2, 3, \ldots, 33\}$.

Let $u = 5$ and $v_1 = 6$, $v_2 = 12$, $v_3 = 17$, $v_4 = 22$, $v_5 = 33$.

We have $o_1 = 6$, $o_2 = 5$, $o_3 = 5$, $o_4 = 11$.

In this case, both $\bar{F}$ and F are maps from k^{33} to k^{27}.

The public key consists of 27 quadratic polynomials with 33 variables. The total number of coefficients for the public key is $27 \times 34 \times 35/2 = 16,065$, or about 15 KB of storage.

The private key consists of 11 polynomials with 22 Vinegar variables and 11 Oil variables, 5 polynomials with 17 Vinegar and 5 Oil variables, 5 polynomials with 12 Vinegar and 5 Oil variables, and 6 polynomials with 6 Vinegar and 6 Oil variables plus the two affine linear transformations L_1 and L_2. The total size is about 10 KB.

This signature scheme signs a document of size $8 \times 27 = 216$ bits with a signature of $8 \times 33 = 264$ bits.

3 Cryptanalysis

We will present a short cryptanalysis of the Rainbow signature scheme by looking at the cryptanalysis of the example above. There are several ways to attack, which we will deal with one by one. For those methods where quadratic forms are used one has to remember that the theory of quadratic forms over finite fields is different when the characteristic is 2 compared to the case when the characteristic is odd [D09].

3.1 Method of Rank Reduction

In [CSV97] a method of rank reduction is used to break the birational permutation signature scheme of Shamir. The reason this attack could work is that the space spanned by the polynomial components of the cipher of Shamir's scheme consists of a flag of spaces:

$$V_1 \subset V_2 \subset \cdots \subset V_t,$$

where V_t is the space spanned by the polynomial components of the cipher, each V_i is a proper subset of V_{i+1} and the rank of the corresponding bilinear form corresponding to the elements in $V_{i+1} - V_i$ is strictly larger than the ones in V_i and the difference of the dimensions between V_i and V_{i+1} is exactly 1. Due to these properties, in particular the last one, it allows one to easily find this flag of spaces, namely all the V_i by first finding V_{n-1} then V_{n-2} and so on by rank reduction.

But this attack method can not work against our scheme anymore. The reason for this is that even though in our case, there also exists such a flag of spaces such that

1) the number of components is exactly the number of layers;
2) the dimension of each component of the flag corresponds exactly to the one of V_{i+1}, $i = 1, \ldots, u-1$;

but

3) the difference in the dimensions of the last two big spaces is exactly O_{u-1}, which we have chosen specifically to be a rather large number 11 unlike in Shamir's case where it is 1.

The property 3) above is exactly the reason why the attack in [CSV97] can not work anymore. The rank reduction method can not be used here due to the fact that $o_{u-1} = 11$ and no longer 1. The "thick last layer of Oil" enables our scheme to resist the rank reduction attack in [CSV97].

3.2 Method of Attack for Oil-Vinegar Schemes

One can see that the action of L_1 is to mix all polynomial components of F. Therefore, each component of the cipher $\bar{F}$ now belongs to the top layer of Oil-Vinegar polynomials, namely they are all elements in P_4. These are Oil and Vinegar polynomials with 22 Vinegar variables and 11 Oil variables.

We can apply the method in [KPG99] for an unbalanced Oil and Vinegar signature scheme in order to try to attack the system, which will allow us to separate the top layer Oil and Vinegar variables. For this, what we need to do is to separate the top (or the final) layer of 11 Oil variables and 22 Vinegar variables. According to the cryptanalysis in [KPG99], the attack complexity of this first step is $q^{22-11-1} \times 11^4 > 2^{90}$.

3.3 Method of Minrank

There are two totally different ways of using the Minrank method. The first one is to search for the polynomial whose associated matrix has the lowest rank among all possible choices. This set of polynomials with 6 Vinegar and 6 Oil variables belongs to the first layer, that is P_1, and was denoted by $\tilde{F}_1$. To do this, we first associate to each polynomial a bilinear form, which has a matrix of size 33×33. We then can use linear combinations of the matrices associated with the components of $\bar{F}$ to derive a polynomial, whose associated matrix has rank 12.

Now, to attack the system, the problem becomes a search for a rank 12 matrix among a group of 27 matrices of size 33×33. From the Minrank method [Cou01] we know that the complexity to find such a matrix is $q^{12} \times 27^3$, which is much larger than 2^{100}.

Another possibility it to search for polynomials corresponding to the polynomials in the second last layer, namely the one that belong to P_3 and come from linear combinations of $\tilde{F}_i$, $i < 4$. In this case, the Minrank method definitely can not be used, because those are of rank 22 in general. One way surely is to randomly search for it. Because the dimension of P_3 is 16, this becomes a problem to search for an element in a subspace of dimension 16 in a total space of dimension 27. Therefore, such a random search needs at least q^{11} searches to find one, but we also need to determine if indeed the rank is lower than 22 for each search. In this case, the total complexity should be at least $q^{11} \times (22 \times 33^2/3) > 2^{100}$. This attack idea is actually related to the attack method in [CSV97], and the argument above explains why the method in [CSV97] can no longer work.

From the most recent e-print results in this direction [WBP], where they study a very general system called STS, we know that their method can also

be applied to our case. In accordance with their estimate, the security of our system is at least $27 \times 33^3 \times (2^8)^{12} \times 5 > 2^{100}$

3.4 The Attack Using the Structure of Multi-layers

For the case of Matsumoto-Imai cryptosystem, Patarin [Pat95] realized that if the cipher is made of several independent parallel "branches", we can perform a separation of variables such that all polynomials in the cipher are derived as linear combinations of polynomials over each group of variables. This property actually can be used to attack the system. At first glance, one would think that our layers look like different "branches". Nevertheless, one should realize that our layers are in no way "independent", because each layer is build upon the previous one. In simple terms one can say that all layers stick together and there is no way we can do any kind of separation of variables. This is clear by looking at the polynomials in the last layer P_4. Therefore the attack using the property of the parallel independent branches in [Pat95] cannot work here. Similarly one can argue that the attack using syszygies cannot work here neither, due to the fact there are no branches and everything is actually "glued together".

3.5 General Methods

Other methods that could be used to attack our signature scheme are those, which solve polynomial equations directly, for example the XL method and its various generalizations, or those, which use Gröbner bases.

Surely, it is very difficult to solve a set of 27 equations with 33 variables, because there are too many solutions for this set of equations. In general, it is much better to solve an equation with only one variable.

Because of the nature of design of our system, one can guess the values for any set of $v_1 = 6$ variables and we have the probability $1/e < 1/2.71828 < 0.37$ to have a unique solution. Now the problem becomes a problem to solve a set of 27 quadratic equations with 33 variables. We should think of it as if it is a set of randomly chosen quadratic equations. According to what is commonly believed, to solve this set of equations, the complexity is at least $2^{3\times 27} > 2^{81}$.

From this we conclude that the total complexity to attack our example is at least 2^{80}.

3.6 General Security Analysis

From the discussion above, we can see that in order to attack the system, one can approach it either from the top layer or form the bottom layer. The security of the bottom layer depends on how effectively the Minrank method can be used. The attack complexity in general is $q^{(v_2-1)}o_{u-1}^3$ if $v_1 > o_1$ or $q^{2v_1}o_{u-1}^3$ if $v_1 \leq o_1$. From this we know that we can not let $v_2 = o_1 + v_1$ be too small.

From the most recent e-print results [WBP], the security of our system is at least $(n - v_1) \times n^3 \times (q)^{o_1+v_1} \times u$, which surely requires $o_1 + v_1$ not to be small.

As for the case of attack from the top, the attack method for unbalanced Oil and Vinegar method tells us that $v_{u-1} - o_{u-1}$ can not be too small. Also to avoid random search attacks o_{u-1} should not be too small.

4 Comparison with Other Multivariable Signature Schemes

In this section, we will present the differences between our new system and two similar multivariable cryptosystems, the unbalanced Oil and Vinegar scheme and the Sflash scheme.

4.1 Comparison with Unbalanced Oil and Vinegar

First, our new system is a generalization of the original Oil and Vinegar construction and the original scheme can be interpreted as just a single layer Rainbow scheme, where $u = 2$.

Let us assume that we want to build an unbalanced Oil and Vinegar scheme, which has the same length for a document that can be signed as our practical example above.

In this case, we choose k again to be a finite field of size $q = 2^8$ and we know that the number of Oil variables should be 27. Because of the attack for unbalance Oil and Vinegar schemes [KPG99], we know that the number of Vinegar variables should be at least $27 + 11 = 38$ in order to have the same level of security.

In this case, the public key consists of 27 polynomials with $38 + 27 = 65$ variables. The size of public key is therefore $27 \times (67 \times 66/2)$ bytes, which is about 116 KB, about 10 times the size of our practical example. This implies that the public computation of verifying the signature will take at least 10 times as long.

The private key for the unbalanced Oil and Vinegar scheme consists of one affine linear transformation on k^{27} and another one on k^{65} and a set of 27 Oil and Vinegar polynomials with 27 Oil variables and 38 Vinegar variables. This means that the private key is about 40 KB. This implies that the private calculation to sign the document will take about four times longer compared to our example.

The length of the signature is $65 \times 8 = 520$ bits, which is also about twice the size of the signature of our example.

From this, we conclude that our scheme should be a much better choice in general in terms of both security and efficiency.

4.2 Sflash

NESSIE, New European Schemes for Signatures, Integrity, and Encryption, is a project within the Information Society Technologies Programme of the European Commission. It made its final selection of the crypto algorithm after a process of more than 2 years. (www.cosic.esat.kuleuven.ac.be/nessie)

Sflash^{v2}, a fast multivariate signature scheme was selected by the Nessie Consortium and was recommended for low-cost smart cards. However, due to security concerns, the designer of Sflash once recommended that Slash^{v2} should not be used, instead a new version Sflash^{v3} is recommended [PGC98]. It is a simple extension of Sflash^{v2} by increasing the length of the signature. Sflash^{v3} has the signature length of 469 bits and a public key of 112 KBytes. But more recently Sflash^{v2} was again deemed to be secure and we compared our implementation to that of Sflash^{v2}.

Sflash^{v2} has a signature of length $37 \times 7 = 259$ for a document of $26 \times 7 = 182$ bits. Our example has a signature of length $33 \times 8 = 264$ for a document of $27 \times 8 = 216$ bits. In terms of per bits efficiency the two are essentially the same.

For a comparison of the running times on a PC, we implemented Sflash^{v_2} as described in [ACDG03]. The generation of the signature is about twice as fast for our example with Rainbow when compared to Sflash. The times for the verification of a signature is of course nearly identical.

From this, we conclude that our scheme should be a good choice in terms of both security and efficiency.

4.3 TTS

We can also compare our system with the new TTS schemes [YC03], but these schemes are broken as was shown in a presentation in IWAP'04 [DY04]. One should also see, that the Tractable Rational Map Signature, as presented in [WHLCY], is very similar to TTS and can be viewed as a very special examples of our scheme.

5 Optimization of Rainbow and Further Generalization

Because of all the possible choices of the design, one has to ask what is the best design. In the practical example above, we presented a very simple realization of Rainbow to make it easier to understand. In this section, we will look at the possibility in general to optimize the scheme for both key size and computational efficiency under the same security requirement.

Let us assume that we want to build a rainbow system to sign a document of size $m \times r$ bits in the space k^n, where k is a finite field of size $q = 2^r$. A question one has to ask is: What is the most efficient choice, if we are given a requirement of a security level of 2^θ?

For a document of length m the length of the signature is $v_u = (m + v_1)$. The security level is determined on the one side by $2^{3r(v_2-1)}$ due the possibility of the Minrank attack. We should choose $v_1 > o_1$ to make the system more efficient, and from this we know that $v_2 = o_1 + v_1$ should be at least $1 + \theta/3r$. But if we want to make the signature as short as possible, the private key as small as possible, and the private calculations as easy as possible, we can see that we should choose v_1 and o_1 such that the difference between o_1 and v_1 should be 0 or 1.

Now assume that we have fixed v_2, o_1, and v_1 already. Due to the security requirement, we know that we should make sure that $q^{v_u - v_{u-1} - 1}(o_{u-1})^4$ is larger than 2^θ.

Let us assume that we have chosen $v_u - v_{u-1}$. The next choice are the layers in-between. Clearly, we can see that the best choice is $v_{i+1} = v_i + 1$, as it has the shortest secret key, the fastest computation speed and it does not affect at all the security of the system. In this case each $\tilde{F}_i$ has only one polynomial.

We suggest to further improve the scheme with an even better choice. For this we set all coefficients of any quadratic term to zero, which mixes the one Oil variable with its Vinegar variables at its layer, and only the coefficient of the linear term of Oil variable is chosen to be a nonzero element. This will ensure that the corresponding linear equation in the signing process always has a solution. It also makes the process faster and does not at all affect the security. We call this type of polynomial, a linear Oil and Vinegar polynomial.

If one wants to make sure to have the maximum probability for success in finding a signature, even the lowest layer should have the same construction, namely $v_2 - v_1 = 1$ and the Oil- and Vinegar polynomial is chosen in the same way. In this case, the only possible place for the signing process will be the top layer. This type of construction, can be viewed also as a combination of the Oil and Vinegar method with the method first suggested in [Sha98].

As for the case of an attack from the top, the attack method for unbalanced Oil and Vinegar method tells us that $v_{u-1} - o_{u-1}$ can not be too small. Also to avoid random search attack o_{u-1} should not be too small.

For example, we can improve our practical example such that $u = 13$ and $v_1 = 6$, $v_2 = 12$, $v_3 = 13$, $v_4 = 14, \ldots, v_{12} = 22$, $v_{13} = 33$, $o_1 = 6$, $o_2 = 1, \ldots, o_{11} = 1$, $o_{12} = 11$. This now is a 12 layer Rainbow scheme.

Another possibility for optimization is to use sparse polynomials when we choose at random the coefficients of the Oil-Vinegar polynomials. Nevertheless, this is a very subtle and delicate task, as it opens up the possibility of new, often hidden and unexpected weakness. The use of sparse polynomial in the new TTS scheme caused it to be broken in [DY04]. Therefore we strongly suggest that such a method should not be used except if one can establish a way to prove that the security level has not been changed.

6 Conclusion

In this paper, we presented a generalization of the Oil and Vinegar signature scheme. It, in general, improves the efficiency of the system. We also suggested to further improve the system by using linear Oil and Vinegar polynomials. We believe that our construction produces excellent multivariable polynomial signature schemes for practical applications.

Acknowledgments

We would like to thank the referees for their helpful comments. The first author also would like to thank the Charles Phelps Taft Research Center for travel support, and Lei Hu, Louis Goubin and Tsuyoshi Takagi for their useful discussions.

References

[ACDG03] Mehdi-Laurent Akkar, Nicolas T. Courtois, Romain Duteuil, and Louis Goubin. A fast and secure implementation of Sflash. In *PKC 2003, LNCS*, volume 2567, pages 267–278. Springer, 2003.

[Cou01] Nicolas T. Courtois. The security of hidden field equations (HFE). In C. Naccache, editor, *Progress in cryptology, CT-RSA, LNCS*, volume 2020, pages 266–281. Springer, 2001.

[CSV97] D. Coppersmith, J. Stern, and S. Vaudenay. The security of the birational permutation signature schemes. *J. Cryptology*, 10(3):207–221, 1997.

[D09] Dickson, Leonard Eugene. Definite forms in a finite field. *Trans. Amer. Math. Soc.*, volume 10, pages 109–122, 1909.

[DY04] Jintai Ding and Z Yin. Cryptanalysis of TTS and Tame–like signature schemes. In *Third International Workshop on Applied Public Key Infrastructures*. Springer, 2004.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *Eurocrypt'99, LNCS*, volume 1592, pages 206–222. Springer, 1999.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In M. Wiener, editor, *Advances in cryptology – Crypto '99, LNCS*, volume 1666, pages 19–30. Springer, 1999.

[MI88] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature verification and message encryption. In C. G. Guenther, editor, *Advances in cryptology – EUROCRYPT '88, LNCS*, volume 330, pages 419–453. Springer, 1988.

[Pat95] J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In D. Coppersmith, editor, *Advances in Cryptology – Crypto '95, LNCS*, volume 963, pages 248–261, 1995.

[Pat96] J. Patarin. Hidden field equations (HFE) and isomorphism of polynomials (IP): Two new families of asymmetric algorithms. In U. Maurer, editor, *Eurocrypt'96, LNCS*, volume 1070, pages 33–48. Springer, 1996.

[PCG01] Jacques Patarin, Nicolas Courtois, and Louis Goubin. Flash, a fast multivariate signature algorithm. In *LNCS*, volume 2020, pages 298–307. Springer, 2001.

[PGC98] Jacques Patarin, Louis Goubin, and Nicolas Courtois. C^*_{-+} and HM: variations around two schemes of T. Matsumoto and H. Imai. In K. Ohta and D. Pei, editors, *ASIACRYPT'98, LNCS*, volume 1514, pages 35–50. Springer, 1998.

[Sha98] Adi Shamir. Efficient signature schemes based on birational permutations. In *LNCS, Advances in cryptology – CRYPTO '98 (Santa Barbara, CA, 1998)*, volume 1462, pages 257–266. Springer, 1998.

[WHLCY] Lih-Chung Wang, Yuh-Hua Hu, Feipei Lai, Chun-Yen Chou, Bo-Yin Yang Tractable Rational Map Signature In Serge Vaudenay, editors, *Public Key Cryptosystems, PKC-2005, LNCS*, volume 3386, pages 244-257 Springer, 2005.

[WBP] Christopher Wolf, An Braeken, and Bart Preneel. Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC. http://eprint.iacr.org/2004/237.

[YC03] B. Yang and J. Chen. A more secure and efficacious TTS signature scheme. *ICISC'03*, 2003. http://eprint.iacr.org/2003/160