

Unbalanced Oil and Vinegar Signature Schemes

- Extended Version -

Aviad Kipnis
NDS Technologies
5 Hamarpe St. Har Hotzvim
Jerusalem - Israel
e-mail : akipnis@ndsisrael.com

Jacques Patarin, Louis Goubin
Bull SmartCards and Terminals
68 route de Versailles - BP 45
78431 Louveciennes Cedex - France
e-mail : {jacques.patarin,louis.goubin}@bull.net

Abstract

In [16], J. Patarin designed a new scheme, called “Oil and Vinegar”, for computing asymmetric signatures. It is very simple, can be computed very fast (both in secret and public key) and requires very little RAM in smartcard implementations. The idea consists in hiding quadratic equations in n unknowns called “oil” and $v = n$ unknowns called “vinegar” over a finite field K , with linear secret functions. This original scheme was broken in [10] by A. Kipnis and A. Shamir. In this paper, we study some very simple variations of the original scheme where $v > n$ (instead of $v = n$). These schemes are called “Unbalanced Oil and Vinegar” (UOV), since we have more “vinegar” unknowns than “oil” unknowns. We show that, when $v \simeq n$, the attack of [10] can be extended, but when $v \geq 2n$ for example, the security of the scheme is still an open problem. Moreover, when $v \simeq \frac{n^2}{2}$, the security of the scheme is exactly equivalent (if we accept a very natural but not proved property) to the problem of solving a random set of n quadratic equations in $\frac{n^2}{2}$ unknowns (with no trapdoor). However, we show that (in characteristic 2) when $v \geq n^2$, finding a solution is generally easy. In this paper, we also present some practical values of the parameters, for which no attacks are known. We also study schemes with public keys of degree three instead of two. We show that no significant advantages exist at the present to recommend schemes of degree three instead of two. However, we show that it is very easy to combine the Oil and Vinegar idea and the HFE schemes of [14]. The resulting scheme, called HFEV, looks at the present also very interesting both from a practical and theoretical point of view. In UOV, the number of vinegar variables must be $> n$, but in HFEV this number can be very small or very large. Then length of a UOV signature can be as short as 192 bits and for HFEV it can be as short as 80 bits.

Note: This paper is the extended version of the paper with the same title published at EUROCRYPT’99.

1 Introduction

Since 1985, various authors (see [7], [9], [12], [14], [16], [17], [18], [21] for example) have suggested some public key schemes where the public key is given as a set of multivariate quadratic (or higher degree) equations over a small finite field K .

The general problem of solving such a set of equations is NP-hard (cf [8]) (even in the quadratic case). Moreover, when the number of unknowns is, say, $n \geq 16$, the best known algorithms are often not significantly better than exhaustive search (when n is very small, Gröbner bases algorithms might be efficient, cf [6]).

The schemes are often very efficient in terms of speed or RAM required in a smartcard implementation. (However, the length of the public key is generally ≥ 1 Kbyte. Nevertheless it is sometimes useful

to notice that secret key computations can be performed without the public key). The most serious problem is that, in order to introduce a trapdoor (to allow the computation of signatures or to allow the decryption of messages when a secret is known), the generated set of public equations generally becomes a small subset of all the possible equations and, in many cases, the algorithms have been broken. For example [7] was broken by their authors, and [12], [16], [21] were broken. However, many schemes are still not broken (for example [14], [17], [18], [20]), and also in many cases, some very simple variations have been suggested in order to repair the schemes. Therefore, at the present, we do not know whether this idea of designing public key algorithms with multivariate polynomials over small finite fields is a very powerful idea (where only some too simple schemes are insecure) or not.

In this paper, we will present two new schemes: UOV and HFEV. UOV is a very simple scheme: the original Oil and Vinegar signature scheme (of [16]) was broken (see [10]), but if we have significantly more “vinegar” unknowns than “oil” unknowns (a definition of the “oil” and “vinegar” unknowns can be found in section 2), then the attack of [10] does not work and the security of this more general scheme (called UOV) is still an open problem.

Moreover, we show that, when we have approximately $\frac{n^2}{2}$ vinegar unknowns for n oil unknowns, the security of UOV is exactly equivalent (if we accept a natural but not proved property) to the problem of solving a random set of n quadratic equations in $\frac{n^2}{2}$ unknowns (with no trapdoor). This result suggests that some partial proof of security (related to some simple to describe and supposed very difficult to solve problems) might be found for some schemes with multivariate polynomials over a finite field. However, we show that most of the systems of n quadratic equations in n^2 (or more) variables can be solved in polynomial complexity... As a result, at the present, we rather recommend $v \simeq 3n$ for example than $v \simeq \frac{n^2}{2}$ for security in UOV. We also study Oil and Vinegar schemes of degree three (instead of two). HFEV combines the ideas of HFE (of [14]) and of vinegar variables. HFEV looks more efficient than the original HFE scheme.

2 The (Original and Unbalanced) Oil and Vinegar of degree two

Let $K = \mathbf{F}_q$ be a small finite field (for example $K = \mathbf{F}_2$). Let n and v be two integers. The message to be signed (or its hash) is represented as an element of K^n , denoted by $y = (y_1, \dots, y_n)$. Typically, $q^n \simeq 2^{128}$. The signature x is represented as an element of K^{n+v} denoted by $x = (x_1, \dots, x_{n+v})$.

Secret key

The secret key is made of two parts:

1. A bijective and affine function $s : K^{n+v} \rightarrow K^{n+v}$. By “affine”, we mean that each component of the output can be written as a polynomial of degree one in the $n + v$ input unknowns, and with coefficients in K .
2. A set (\mathcal{S}) of n equations of the following type:

$$\forall i, 1 \leq i \leq n, y_i = \sum \gamma_{ijk} a_j a'_k + \sum \lambda_{ijk} a'_j a'_k + \sum \xi_{ij} a_j + \sum \xi'_{ij} a'_j + \delta_i \quad (\mathcal{S}).$$

The coefficients γ_{ijk} , λ_{ijk} , ξ_{ij} , ξ'_{ij} and δ_i are the secret coefficients of these n equations. The values a_1, \dots, a_n (the “oil” unknowns) and a'_1, \dots, a'_v (the “vinegar” unknowns) lie in K . Note that these equations (\mathcal{S}) contain no terms in $a_i a_j$.

Public key

Let A be the element of K^{n+v} defined by $A = (a_1, \dots, a_n, a'_1, \dots, a'_v)$. A is transformed into $x = s^{-1}(A)$, where s is the secret, bijective and affine function from K^{n+v} to K^{n+v} .

Each value y_i , $1 \leq i \leq n$, can be written as a polynomial P_i of total degree two in the x_j unknowns, $1 \leq j \leq n + v$. We denote by (\mathcal{P}) the set of these n equations:

$$\forall i, 1 \leq i \leq n, y_i = P_i(x_1, \dots, x_{n+v}) \quad (\mathcal{P}).$$

These n quadratic equations (\mathcal{P}) (in the $n + v$ unknowns x_j) are the public key.

Computation of a signature (with the secret key)

The computation of a signature x of y is performed as follows:

Step 1: We find n unknowns a_1, \dots, a_n of K and v unknowns a'_1, \dots, a'_v of K such that the n equations (\mathcal{S}) are satisfied.

This can be done as follows: we randomly choose the v vinegar unknowns a'_i , and then we compute the a_i unknowns from (\mathcal{S}) by Gaussian reductions (because – since there are no $a_i a_j$ terms – the (\mathcal{S}) equations are affine in the a_i unknowns when the a'_i are fixed).

Remark: If we find no solution, then we simply try again with new random vinegar unknowns. After very few tries, the probability of obtaining at least one solution is very high, because the probability for a $n \times n$ matrix over \mathbf{F}_q to be invertible is not negligible. (It is exactly $(1 - \frac{1}{q})(1 - \frac{1}{q^2}) \dots (1 - \frac{1}{q^{n-1}})$. For $q = 2$, this gives approximately 30 %, and for $q > 2$, this probability is even larger.)

Step 2: We compute $x = s^{-1}(A)$, where $A = (a_1, \dots, a_n, a'_1, \dots, a'_v)$. x is a signature of y .

Public verification of a signature

A signature x of y is valid if and only if all the (\mathcal{P}) are satisfied. As a result, no secret is needed to check whether a signature is valid: this is an asymmetric signature scheme.

Note: The name “Oil and Vinegar” comes from the fact that – in the equations (\mathcal{S}) – the “oil unknowns” a_i and the “vinegar unknowns” a'_j are not all mixed together: there are no $a_i a_j$ products. However, in (\mathcal{P}) , this property is hidden by the “mixing” of the unknowns by the s transformation. Is this property “hidden enough”? In fact, this question exactly means: “is the scheme secure?”. When $v = n$, we call the scheme “Original Oil and Vinegar”, since this case was first presented in [16]. This case was broken in [10]. It is very easy to see that the cryptanalysis of [10] also works, exactly in the same way, when $v < n$. However, the cases $v > n$ are, as we will see, much more difficult. When $v > n$, we call the scheme “Unbalanced Oil and Vinegar”.

3 A short description of the attack of [10]: cryptanalysis of the case

$$v = n$$

The idea of the attack of [10] is essentially the following:

In order to separate the oil variables and the vinegar variables, we look at the quadratic forms of the n public equations of (\mathcal{P}) , we omit for a while the linear terms. Let G_i for $1 \leq i \leq n$ be the respective matrix of the quadratic form of P_i of the public equations (\mathcal{P}) .

The quadratic part of the equations in the set (\mathcal{S}) is represented as a quadratic form with a corresponding $2n \times 2n$ matrix of the form : $\begin{pmatrix} 0 & A \\ B & C \end{pmatrix}$, the upper left $n \times n$ zero submatrix is due to the fact that an oil variable is not multiplied by an oil variable.

After hiding the internal variables with the linear function s , we get a representation for the matrices $G_i = S \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} S^t$, where S is an invertible $2n \times 2n$ matrix.

Definition 3.1: We define the oil subspace to be the linear subspace of all vectors in K^{2n} whose second half contains only zeros.

Definition 3.2: We define the vinegar subspace as the linear subspace of all vectors in K^{2n} whose first half contains only zeros.

Lemma 1 *Let E and F be a $2n \times 2n$ matrices with an upper left zero $n \times n$ submatrix. If F is invertible then the oil subspace is an invariant subspace of EF^{-1} .*

Proof: E and F map the oil subspace into the vinegar subspace. If F is invertible, then this mapping between the oil subspace and the vinegar subspace is one to one and onto (here we use the assumption that $v = n$). Therefore F^{-1} maps back the vinegar subspace into the oil subspace this argument explains why the oil subspace is transformed into itself by EF^{-1} .

Definition 3.4: For an invertible matrix G_j , define $G_{ij} = G_i G_j^{-1}$.

Definition 3.5: Let O be the image of the oil subspace by S^{-1} . In order to find the oil subspace, we use the following theorem:

Theorem 3.1 O is a common invariant subspace of all the matrices G_{ij} .

Proof:

$$G_i G_j^{-1} = S \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} S^t (S^t)^{-1} \begin{pmatrix} 0 & A_j \\ B_j & C_j \end{pmatrix}^{-1} S^{-1} = S \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} \begin{pmatrix} 0 & A_j \\ B_j & C_j \end{pmatrix}^{-1} S^{-1}$$

The two inner matrices have the form of E and F in lemma 1. Therefore, the oil subspace is an invariant subspace of the inner term and O is an invariant subspace of $G_i G_j^{-1}$.

The problem of finding common invariant subspace of set of matrices is studied in [10]. Applying the algorithms in [10] gives us O . We then pick V to be an arbitrary subspace of dimension n such that $V + O = K^{2n}$, and they give an equivalent oil and vinegar separation.

Once we have such a separation, we bring back the linear terms that were omitted, we pick random values for the vinegar variables and left with a set of n linear equations with n oil variables.

Note: Lemma 1 is not true any more when $v > n$. The oil subspace is still mapped by E and F into the vinegar subspace. However F^{-1} does not necessary maps the image by E of the oil subspace back into the oil subspace and this is why the cryptanalysis of the original oil and vinegar is not valid for the unbalanced case.

This corresponds to the fact that, if the submatrix of zeros in the top left corner of F is smaller than $n \times n$, then F^{-1} does not have (in general) a submatrix of zeros in the bottom right corner. For example:

$$\begin{pmatrix} 0 & 3 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \end{pmatrix}^{-1} = \frac{1}{3} \begin{pmatrix} 2 & -5 & 4 \\ 2 & -2 & 1 \\ -3 & 6 & -3 \end{pmatrix}.$$

However, when $v - n$ is small, we see in the next section how to extend the attack.

4 Cryptanalysis when $v > n$ and $v \simeq n$

In this section, we discuss the case of Oil and Vinegar schemes where $v > n$, although a direct application of the attack described in [10] and in the previous section does not solve the problem, a modification of the attack exists, that is applicable as long as $v - n$ is small (more precisely the expected complexity of the attack is approximately $q^{(v-n)-1} \cdot n^4$).

Definition 4.1: We define in this section the oil subspace to be the linear subspace of all vectors in K^{n+v} whose last v coordinates are only zeros.

Definition 4.2: We define in this section the vinegar subspace to be the linear subspace of all vectors in K^{n+v} whose first n coordinates are only zeros.

Here in this section, we start with the homogeneous quadratic terms of the equations: we omit the linear terms for a while.

The matrices G_i have the representation

$$G_i = S \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} S^t$$

where the upper left matrix is the $n \times n$ zero matrix, A_i is a $n \times v$ matrix, B_i is a $v \times n$ matrix, C_i is a $v \times v$ matrix and S is a $(n + v) \times (n + v)$ invertible linear matrix.

Definition 4.3: Define E_i to be $\begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix}$.

Lemma 2 For any matrix E that has the form $\begin{pmatrix} 0 & A \\ B & C \end{pmatrix}$, the following holds:

- a) E transforms the oil subspace into the vinegar subspace.
- b) If the matrix E^{-1} exists, then the image of the vinegar subspace by E^{-1} is a subspace of dimension v which contains the n -dimensional oil subspace in it.

Proof: a) follows directly from the definition of the oil and vinegar subspaces. When a) is given then b) is immediate.

The algorithm we propose is a probabilistic algorithm. It looks for an invariant subspace of the oil subspace after it is transformed by S . The probability for the algorithm to succeed on the first try is small. Therefore we need to repeat it with different inputs. We use the following property: any linear combination of the matrices E_1, \dots, E_n is also of the form $\begin{pmatrix} 0 & A \\ B & C \end{pmatrix}$.

The following theorem explains why an invariant subspace may exist with a certain probability.

Theorem 4.1 Let F be an invertible linear combination of the matrices E_1, \dots, E_n . Then for any k such that E_k^{-1} exists, the matrix FE_k^{-1} has a non trivial invariant subspace which is also a subspace of the oil subspace, with probability not less than $\frac{q-1}{q^{2d}-1}$ for $d = v - n$.

Proof: The matrix F maps the oil subspace into the vinegar subspace, the image by F of the oil subspace is mapped by E_k^{-1} into a subspace of dimension v that contains the oil subspace – these are due to lemma 1. We write $v = n + d$, where d is a small integer. The oil subspace and its image by FE_k^{-1} are two subspaces with dimension n that reside in a subspace of dimension $n + d$. Therefore, their intersection is a subspace of the oil subspace with dimension not less than $n - d$. We denote the oil subspace by I_0 and the intersection subspace by I_1 . Now, we take the inverse images by FE_k^{-1} of I_1 : this is a subspace of I_0 (the oil subspace) with dimension not less than $n - d$ and the intersection between this subspace and I_1 is a subspace of I_1 with dimension not less than $n - 2d$. We call this subspace I_2 . We can continue this process and define I_ℓ to be the intersection of $I_{\ell-1}$ and its inverse image by FE_k^{-1} . These two subspaces have co-dimension not more than d in $I_{\ell-2}$. Therefore, I_ℓ has a co-dimension not more than $2d$ in $I_{\ell-2}$ or a co-dimension not more than d in $I_{\ell-1}$. We can carry on this process as long as we are sure that the inverse image by FE_k^{-1} of I_ℓ has a non trivial intersection with I_ℓ . This is ensured as long as the dimension of I_ℓ is greater than d , but when the dimension is d or less than d , there is no guaranty that these two subspaces – that reside in $I_{\ell-1}$ – have a non trivial intersection. We end the process with I_ℓ that has dimension $\leq d$ that resides in $I_{\ell-1}$ with dimension not more than $2d$.

We know that the transformation $(EG_k^{-1})^{-1}$ maps I_ℓ into $I_{\ell-1}$. With probability not less than $\frac{q-1}{q^{2d}-1}$, there is a non zero vector in I_ℓ that is mapped to a non zero multiple of itself – and therefore there is a non trivial subspace of FE_k^{-1} which is also a subspace of the oil subspace.

Note: It is possible to get a better result for the expected number of eigenvectors and with much less effort: I_1 is a subspace with dimension not less than $n - d$ and is mapped by FE_k^{-1} into a subspace with dimension n . The probability for a non zero vector to be mapped to a non zero multiple of itself is $\frac{q-1}{q^n-1}$. To get the expected value, we multiply it by the number of non zero vectors in I_1 . It gives a value which is not less than $\frac{(q-1)(q^{n-d}-1)}{q^n-1}$. Since every eigenvector is counted $q - 1$ times, then the expected number of invariant subspaces of dimension 1 is not less than $\frac{q^{n-d}-1}{q^n-1} \sim q^{-d}$.

We define O as in section 3 and we get the following result for O :

Theorem 4.2 *Let F be an invertible linear combination of the matrices $G_1; \dots, G_n$. Then for any k such that G_k^{-1} exists, the matrix FG_k^{-1} has a non trivial invariant subspace, which is also a subspace of O with probability not less than $\frac{q-1}{q^{2d}-1}$ for $d = v - n$.*

Proof:

$$\begin{aligned} FG_k^{-1} &= (\alpha_1 G_1 + \dots + \alpha_n G_n) G_k^{-1} \\ &= S(\alpha_1 E_1 + \dots + \alpha_n E_n) S^t (S^t)^{-1} E_k^{-1} S^{-1} = S(\alpha_1 E_1 + \dots + \alpha_n E_n) E_k^{-1} S^{-1}. \end{aligned}$$

The inner term is an invariant subspace of the oil subspace with the required probability. Therefore, the same will hold for FG_k^{-1} , but instead of a subspace of the oil subspace, we get a subspace of O .

How to find O ?

We take a random linear combination of G_1, \dots, G_n and multiply it by an inverse of one of the G_k matrices. Then we calculate all the minimal invariant subspaces of this matrix (a minimal invariant subspace of a matrix A contains no non trivial invariant subspaces of the matrix A – these subspaces corresponds to irreducible factors of the characteristic polynomial of A). This can be done in probabilistic polynomial time using standard linear algebra techniques. This matrix may have an invariant subspace which is a subspace of O .

The following lemma enables us to distinguish between subspaces that are contained in O and random subspaces.

Lemma 3 *If H is a linear subspace and $H \subset O$, then for every x, y in H and every i , $G_i(x, y) = 0$ (here we regard G_i as a bilinear form).*

Proof: There are x' and y' in the oil subspace such that $x' = xS^{-1}$ and $y' = yS^{-1}$.

$$Gi(x, y) = xS \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} S^t y^t = (x'S^{-1})S \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} ((y'S^{-1})S)^t = x' \begin{pmatrix} 0 & A_i \\ B_i & C_i \end{pmatrix} (y')^t = 0.$$

The last term is zero because x' and y' are in the oil subspace.

This lemma gives a polynomial test to distinguish between subspaces of O and random subspaces.

If the matrix we used has no minimal subspace which is also a subspace of O , then we pick another linear combination of G_1, \dots, G_n , multiply it by an inverse of one of the G_k matrices and try again.

After repeating this process approximately q^{d-1} times, we find with good probability at least one zero vector of O . We continue the process until we get n independent vectors of O . These vectors span O . The expected complexity of the process is proportional to $q^{d-1} \cdot n^4$. We use here the expected number of tries until we find a non trivial invariant subspace and the term n^4 covers the computational linear algebra operations we need to perform for every try.

5 The cases $v \simeq \frac{n^2}{2}$ (or $v \geq \frac{n^2}{2}$)

Property

Let (\mathcal{A}) be a random set of n quadratic equations in $(n + v)$ variables x_1, \dots, x_{n+v} . (By “random” we mean that the coefficients of these equations are uniformly and randomly chosen). When $v \simeq \frac{n^2}{2}$ (and more generally when $v \geq \frac{n^2}{2}$), there is probably – for most of such (\mathcal{A}) – a linear change of variables $(x_1, \dots, x_{n+v}) \mapsto (x'_1, \dots, x'_{n+v})$ such that the set (\mathcal{A}') of (\mathcal{A}) equations written in (x'_1, \dots, x'_{n+v}) is an “Oil and Vinegar” system (i.e. there are no terms in $x'_i \cdot x'_j$ with $i \leq n$ and $j \leq n$).

An argument to justify the property

Let

$$\begin{cases} x_1 = \alpha_{1,1}x'_1 + \alpha_{1,2}x'_2 + \dots + \alpha_{1,n+v}x'_{n+v} \\ \vdots \\ x_{n+v} = \alpha_{n+v,1}x'_1 + \alpha_{n+v,2}x'_2 + \dots + \alpha_{n+v,n+v}x'_{n+v} \end{cases}$$

By writing that the coefficient in all the n equations of (\mathcal{A}) of all the $x'_i \cdot x'_j$ ($i \leq n$ and $j \leq n$) is zero, we obtain a system of $n \cdot n \cdot \frac{n+1}{2}$ quadratic equations in the $(n+v) \cdot n$ variables $\alpha_{i,j}$ ($1 \leq i \leq n+v$, $1 \leq j \leq n$). Therefore, when $v \geq$ approximately $\frac{n^2}{2}$, we may expect to have a solution for this system of equations for most of (\mathcal{A}) .

Remarks:

1. This argument is very natural, but this is not a complete mathematical proof.
2. The system may have a solution, but finding the solution might be a difficult problem. This is why an Unbalanced Oil and Vinegar scheme might be secure (for well chosen parameters): there is always a linear change of variables that makes the problem easy to solve, but finding such a change of variables might be difficult.
3. In section 7, we will see that, despite the result of this section, it is not recommended to choose $v \geq n^2$.

6 Solving a set of n quadratic equations in k unknowns, $k > n$, is NP-hard

We present in section 7 an algorithm that solves in polynomial complexity more than 99% of the sets of n quadratic equations in n^2 (or more) variables (i.e. it will probably succeed in more than 99% of the cases when the coefficients are randomly chosen).

Roughly speaking, we can summarize this result by saying that solving a “random” set of n quadratic equations in n^2 (or more) variables is feasible in polynomial complexity (and thus is not NP-hard if $P \neq NP$). However, we see in the present section that the problem of solving **any** (i.e. 100%) set of n quadratic equations in $k \geq n$ variables (so for example in $k = n^2$ variables) is NP-hard !

To see this, let us assume that we have a black box that takes any set of n quadratic equations with k variables in input, and that gives one solution when at least one solution exists. Then we can use this black box to find a solution for any set of n quadratic equations in n variables (and this is NP-hard). We proceed (for example) as follows. Let (\mathcal{A}) be a set of $(n-1)$ quadratic equations with $(n-1)$ variables x_1, x_2, \dots, x_{n-1} . Then let y_1, \dots, y_α be α more variables.

Let (\mathcal{B}) be the set of (\mathcal{A}) equations plus one quadratic equation in y_1, \dots, y_α (for example the equation: $(y_1 + \dots + y_\alpha)^2 = 1$). Then (\mathcal{B}) is a set of exactly n quadratic equations in $(n+1+\alpha)$ variables. It is clear that from the solution of (\mathcal{B}) we will immediately find one solution for (\mathcal{A}) .

Note 1: (\mathcal{B}) has a very special shape ! This is why there is a polynomial algorithm for 99% of the equations without contradicting the fact that solving these sets (\mathcal{B}) of equations is a NP-hard problem.

Note 2: For (\mathcal{B}) , we can also add more than one quadratic equations in the y_i variables and we can linearly mix these equations with the equations of (\mathcal{A}) . In this case, (\mathcal{B}) is still of very special form but this very special form is less obvious at first glance since all the variables x_i and y_j are in all the equations of (\mathcal{B}) .

7 A generally efficient algorithm for solving a random set of n quadratic equations in n^2 (or more) unknowns

In this section, we describe an algorithm that solves a system of n randomly chosen quadratic equations in $n+v$ variables, when $v \geq n^2$.

Let (\mathcal{S}) be the following system:

$$(\mathcal{S}) \quad \begin{cases} \sum_{1 \leq i \leq j \leq n+v} a_{ij1} x_i x_j + \sum_{1 \leq i \leq n+v} b_{i1} x_i + \delta_1 = 0 \\ \vdots \\ \sum_{1 \leq i \leq j \leq n+v} a_{ijn} x_i x_j + \sum_{1 \leq i \leq n+v} b_{in} x_i + \delta_n = 0 \end{cases}$$

The main idea of the algorithm consists in using a change of variables such as:

$$\begin{cases} x_1 = \alpha_{1,1} y_1 + \alpha_{2,1} y_2 + \dots + \alpha_{n,1} y_n + \alpha_{n+1,1} y_{n+1} + \dots + \alpha_{n+v,1} y_{n+v} \\ \vdots \\ x_{n+v} = \alpha_{1,n+v} y_1 + \alpha_{2,n+v} y_2 + \dots + \alpha_{n,n+v} y_n + \alpha_{n+1,n+v} y_{n+1} + \dots + \alpha_{n+v,n+v} y_{n+v} \end{cases}$$

whose $\alpha_{i,j}$ coefficients (for $1 \leq i \leq n$, $1 \leq j \leq n+v$) are found step by step, in order that the resulting system (\mathcal{S}') (written with respect to these new variables y_1, \dots, y_{n+v}) is easy to solve.

- We begin by choosing randomly $\alpha_{1,1}, \dots, \alpha_{1,n+v}$.
- We then compute $\alpha_{2,1}, \dots, \alpha_{2,n+v}$ such that (\mathcal{S}') contains no $y_1 y_2$ terms. This condition leads to a system of n linear equations on the $(n+v)$ unknowns $\alpha_{2,j}$ ($1 \leq j \leq n+v$):

$$\sum_{1 \leq i \leq j \leq n+v} a_{ijk} \alpha_{1,i} \alpha_{2,j} = 0 \quad (1 \leq k \leq n).$$

- We then compute $\alpha_{3,1}, \dots, \alpha_{3,n+v}$ such that (\mathcal{S}') contains neither $y_1 y_3$ terms, nor $y_2 y_3$ terms. This condition is equivalent to the following system of $2n$ linear equations on the $(n+v)$ unknowns $\alpha_{3,j}$ ($1 \leq j \leq n+v$):

$$\begin{cases} \sum_{1 \leq i \leq j \leq n+v} a_{ijk} \alpha_{1,i} \alpha_{3,j} = 0 & (1 \leq k \leq n) \\ \sum_{1 \leq i \leq j \leq n+v} a_{ijk} \alpha_{2,i} \alpha_{3,j} = 0 & (1 \leq k \leq n) \end{cases}$$

- ...
- Finally, we compute $\alpha_{n,1}, \dots, \alpha_{n,n+v}$ such that (\mathcal{S}') contains neither $y_1 y_n$ terms, nor $y_2 y_n$ terms, ..., nor $y_{n-1} y_n$ terms. This condition gives the following system of $(n-1)n$ linear equations on the $(n+v)$ unknowns $\alpha_{n,j}$ ($1 \leq j \leq n+v$):

$$\begin{cases} \sum_{1 \leq i \leq j \leq n+v} a_{ijk} \alpha_{1,i} \alpha_{n,j} = 0 & (1 \leq k \leq n) \\ \vdots \\ \sum_{1 \leq i \leq j \leq n+v} a_{ijk} \alpha_{n-1,i} \alpha_{n,j} = 0 & (1 \leq k \leq n) \end{cases}$$

In general, all these linear equations provide at least one solution (found by Gaussian reductions). In particular, the last system of $n(n-1)$ equations and $(n+v)$ unknowns generally gives a solution, as soon as $n+v > n(n-1)$, i.e. $v > n(n-2)$, which is true by hypothesis.

Moreover, the n vectors $\begin{pmatrix} \alpha_{1,1} \\ \vdots \\ \alpha_{1,n+v} \end{pmatrix}, \dots, \begin{pmatrix} \alpha_{n,1} \\ \vdots \\ \alpha_{n,n+v} \end{pmatrix}$ are very likely to be linearly independent for a

random quadratic system (\mathcal{S}) .

The remaining $\alpha_{i,j}$ constants (i.e. those with $n+1 \leq i \leq n+v$ and $1 \leq j \leq n+1$) are randomly chosen, so as to obtain a *bijective* change of variables.

By rewriting the system (\mathcal{S}) with respect to these new variables y_i , we are led to the following system:

$$(\mathcal{S}') \quad \begin{cases} \sum_{i=1}^n \beta_{i,1} y_i^2 + y_1 L_{1,1}(y_{n+1}, \dots, y_{n+v}) + \dots + y_n L_{n,1}(y_{n+1}, \dots, y_{n+v}) + Q_1(y_{n+1}, \dots, y_{n+v}) = 0 \\ \vdots \\ \sum_{i=1}^n \beta_{i,n} y_i^2 + y_1 L_{1,n}(y_{n+1}, \dots, y_{n+v}) + \dots + y_n L_{n,n}(y_{n+1}, \dots, y_{n+v}) + Q_n(y_{n+1}, \dots, y_{n+v}) = 0 \end{cases}$$

where each $L_{i,j}$ is an affine function and each Q_i is a quadratic function.
 We then compute y_{n+1}, \dots, y_{n+v} such that:

$$\forall i, 1 \leq i \leq n, \forall j, 1 \leq j \leq n + v, L_{i,j}(y_{n+1}, \dots, y_{n+v}) = 0.$$

This is possible because we have to solve a linear system of n^2 equations and v unknowns, which generally provides at least one solution, as long as $v \geq n^2$. We pick one of these solutions.

It remains to solve the following system of n equations on the n unknowns y_1, \dots, y_n :

$$(S'') \quad \begin{cases} \sum_{i=1}^n \beta_{i1} y_i^2 = \lambda_1 \\ \vdots \\ \sum_{i=1}^n \beta_{in} y_i^2 = \lambda_n \end{cases}$$

where $\lambda_k = -Q_k(y_{n+1}, \dots, y_{n+v})$ ($1 \leq k \leq n$).

In general, this gives the y_i^2 by Gaussian reduction.

Then, in characteristic 2, since $x \mapsto x^2$ is a bijection, we will then find a solution for the y_i from this expression of the y_i^2 .

Note: In characteristic $\neq 2$, this algorithm will also succeed when 2^n is not too large (*i.e.* when $n \leq 40$ for example). (However, when $2^n \geq 2^{64}$ and when the characteristic is $\neq 2$, this algorithm requires too many computations.)

8 A variation with twice smaller signatures

In the UOV described in section 2, the public key is a set of n quadratic equations $y_i = P_i(x_1, \dots, x_{n+v})$, for $1 \leq i \leq n$, where $y = (y_1, \dots, y_n)$ is the hash value of the message to be signed. If we use a collision-free hash function, the hash value must at least be 128 bits long. Therefore, q^n must be at least 2^{128} , so that the typical length of the signature, if $v = 2n$, is at least $3 \times 128 = 384$ bits.

As we see now, it is possible to make a small variation in the signature design in order to obtain twice smaller signatures. The idea is to keep the same polynomial P_i (with the same associated secret key), but now the public equations that we check are:

$$\forall i, P_i(x_1, \dots, x_{n+v}) + L_i(y_1, \dots, y_n, x_1, \dots, x_{n+v}) = 0,$$

where L_i is a linear function in (x_1, \dots, x_{n+v}) and where the coefficients of L_i are generated by a hash function in (y_1, \dots, y_n) .

For example $L_i(y_1, \dots, y_n, x_1, \dots, x_{n+v}) = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_{n+v} x_{n+v}$, where $(\alpha_1, \alpha_2, \dots, \alpha_{n+v}) = \text{Hash}(y_1, \dots, y_n || i)$. Now, n can be chosen such that $q^n \geq 2^{64}$ (instead $q^n \geq 2^{128}$). (Note: q^n must be $\geq 2^{64}$ in order to avoid exhaustive search on a solution x). If $v = 2n$ and $q^n \simeq 2^{64}$, the length of the signature will be $3 \times 64 = 192$ bits.

9 Oil and Vinegar of degree three

9.1 The scheme

The quadratic Oil and Vinegar schemes described in section 2 can easily be extended to any higher degree. We now present the schemes in degree three.

Variables

Let K be a small finite field (for example $K = \mathbf{F}_2$). Let a_1, \dots, a_n be n elements of K , called the “oil” unknowns. Let a'_1, \dots, a'_v be v elements of K , called the “vinegar” unknowns.

Secret key.

The secret key is made of two parts:

1. A bijective and affine function $s : K^{n+v} \rightarrow K^{n+v}$.
2. A set (\mathcal{S}) of n equations of the following type: for all $i \leq n$,

$$y_i = \sum \gamma_{ijk\ell} a_j a'_k a'_\ell + \sum \mu_{ijk\ell} a'_j a'_k a'_\ell + \sum \lambda_{ijk} a_j a'_k + \sum \nu_{ijk} a'_j a'_k + \sum \xi_{ij} a_j + \sum \xi'_{ij} a'_j + \delta_i \quad (\mathcal{S}).$$

The coefficients $\gamma_{ijk\ell}$, $\mu_{ijk\ell}$, λ_{ijk} , ν_{ijk} , ξ_{ij} , ξ'_{ij} and δ_i are the secret coefficients of these n equations. Note that these equations (\mathcal{S}) contain no terms in $a_j a_k a_\ell$ or in $a_j a_k$: the equations are affine in the a_j unknowns when the a'_k unknowns are fixed.

Public key

Let A be the element of K^{n+v} defined by $A = (a_1, \dots, a_n, a'_1, \dots, a'_v)$. A is transformed into $x = s^{-1}(A)$, where s is the secret, bijective and affine function from K^{n+v} to K^{n+v} . Each value y_i , $1 \leq i \leq n$, can be written as a polynomial P_i of total degree three in the x_j unknowns, $1 \leq j \leq n+v$. We denote by (\mathcal{P}) the set of the following n equations:

$$\forall i, 1 \leq i \leq n, y_i = P_i(x_1, \dots, x_{n+v}) \quad (\mathcal{P}).$$

These n equations (\mathcal{P}) are the public key.

Computation of a signature

Let y be the message to be signed (or its hash value).

Step 1: We randomly choose the v vinegar unknowns a'_i , and then we compute the a_i unknowns from (\mathcal{S}) by Gaussian reductions (because – since there are no $a_i a_j$ terms – the (\mathcal{S}) equations are affine in the a_i unknowns when the a'_i are fixed. (If we find no solution for this affine system of n equations and n “oil” unknowns, we just try again with new random “vinegar” unknowns.)

Step 2: We compute $x = s^{-1}(A)$, where $A = (a_1, \dots, a_n, a'_1, \dots, a'_v)$. x is a signature of y .

Public verification of a signature

A signature x of y is valid if and only if all the (\mathcal{P}) are satisfied.

9.2 First cryptanalysis of Oil and Vinegar of degree three when $v \leq n$

We can look at the quadratic part of the public key and attack it exactly as for an Oil and Vinegar of degree two. This is expected to work when $v \leq n$.

Note: If there is no quadratic part (i.e. is the public key is homogeneous of degree three), or if this attack does not work, then it is always possible to apply a random affine change of variables and to try again. Moreover, we will see in section 9.3 that, surprisingly, there is an even easier and more efficient attack in degree three than in degree two !

9.3 Cryptanalysis of Oil and Vinegar of degree three when $v \leq (1 + \sqrt{3})n$ and K is of characteristic $\neq 2$ (from an idea of [4])

The key idea is to detect a “linearity” in some directions. We search the set V of the values $d = (d_1, \dots, d_{n+v})$ such that:

$$\forall x, \forall i, 1 \leq i \leq n, P_i(x + d) + P_i(x - d) = 2P_i(x) \quad (\#).$$

By writing that each x_k indeterminate has a zero coefficient, we obtain $n \cdot (n + v)$ quadratic equations in the $(n + v)$ unknowns d_j .

(Each monomial $x_i x_j x_k$ gives $(x_j + d_j)(x_k + d_k)(x_\ell + d_\ell) + (x_j - d_j)(x_k - d_k)(x_\ell - d_\ell) - 2x_j x_k x_\ell$, i.e. $2(x_j d_k d_\ell + x_k d_j d_\ell + x_\ell d_j d_k)$.)

Furthermore, the cryptanalyst can specify about $n - 1$ of the coordinates d_k of d , since the vectorial space of the correct d is of dimension n . It remains thus to solve $n \cdot (n + v)$ quadratic equations in $(v + 1)$ unknowns d_j . When v is not too large (typically when $\frac{(v+1)^2}{2} \leq n(n + v)$, i.e. when $v \leq (1 + \sqrt{3})n$), this is expected to be easy.

As a result when $v \leq$ approximately $(1 + \sqrt{3})n$ and $|K|$ is odd, this gives a simple way to break the scheme.

Note 1: When v is sensibly greater than $(1 + \sqrt{3})n$ (this is a more unbalanced limit than what we had in the quadratic case), we do not know at the present how to break the scheme.

Note 2: Strangely enough, this cryptanalysis of degree three Oil and Vinegar schemes does not work on degree two Oil and Vinegar schemes. The reason is that – in degree two – writing

$$\forall x, \forall i, 1 \leq i \leq n, P_i(x + d) + P_i(x - d) = 2P_i(x)$$

only gives n equations of degree two on the $(n + v)$ d_j unknowns (that we do not know how to solve). (Each monomial $x_j x_k$ gives $(x_j + d_j)(x_k + d_k) + (x_j - d_j)(x_k - d_k) - 2x_j x_k$, i.e. $2d_j d_k$.)

Note 3: In degree two, we have seen that Unbalanced Oil and Vinegar public keys are expected to cover almost all the set of n quadratic equations when $v \simeq \frac{n^2}{2}$. In degree three, we have a similar property: the public keys are expected to cover almost all the set of n cubic equations when $v \simeq \frac{n^3}{6}$ (the proof is similar).

10 Public key length

If we choose $K = \mathbf{F}_2$ then the public key is often large. So it is often more practical to choose a larger K and a smaller n : then the length of the public key can be reduced a lot (see the examples in section 14). However, even when K and n are fixed, it is always feasible to make some easy transformations on a public key in order to obtain the public key in a canonical way such that this canonical expression is slightly shorter than the original expression.

- First, it is always possible to publish only the homogeneous part of the quadratic equations (and not the linear part), because if we know the secret affine change of variables in an Oil and Vinegar scheme with a public key P , then we can solve $P(x) + L(x) = y$, where L is any linear expression with exactly the same affine change of variables. It is thus possible to publish only the homogeneous part of P and to choose a convention for computing the linear part L of the public key (instead of publishing L). For example, this convention can be that the linear terms of L in the equation number i ($1 \leq i \leq n$) are computed from $\text{Hash}(i||Id)$ (or from $\text{Hash}(i||P)$), where Hash is a public hash function and where Id is the identity of the owner of the secret key.

Remark: It is also possible to decide that the linear part is always zero. However, from a theoretical point of view, this may be less secure because we cannot exclude the possibility that some efficient attacks exist against the homogeneous Oil and Vinegar without finding the secret key (and without breaking the non-homogeneous case).

- On the equations, it is also possible to:
 1. Make linear and bijective changes of variable $x' = A(x)$.
 2. Compute a linear and bijective transformation on the equation: $\mathcal{P}' = t(\mathcal{P})$. (For example, the new first equation can be the old first plus the old third equation, etc).

By combining easily these two transformations, it is always possible to decrease slightly the length of the public key.

Idea 1: It is possible to make a change of variables such that the first equation is in a canonical form (see [11], chapter 6). With this presentation of the public key, the length of the public key will be approximately $\frac{n-1}{n}$ times the initial length.

Idea 2: Another idea is to use the idea of section 7, i.e. to create a square of $\lambda \times \lambda$ zeros in the coefficients, where $\lambda \simeq \sqrt{n+v}$. With this presentation, the length of the public key is approximately $\frac{(n+v)^2 - (n+v)}{(n+v)^2}$ times the initial length.

Remark: As we will see in section 13, the most efficient way of reducing the length of the public key is to choose carefully the values q and n .

11 Another variation of the schemes: Unbalanced Oil, Vinegar and Salt

The scheme

Let (\mathcal{A}) be a set of n quadratic “Oil and Vinegar” equations, as described above, with n oil variables and v vinegar variables. We denote by (q_1, \dots, q_n) these equations. Let (\mathcal{A}') be a set of r truly random quadratic equations in all the variables (i.e. we can have terms in $a_i a_j$ where a_i and a_j are oil variables in (\mathcal{A}') but not in (\mathcal{A})). We denote by (q'_1, \dots, q'_r) these equations. We will call these r equations the “salt” equations.

Let t be a secret affine permutation of $K^{n+r} \rightarrow K^{n+r}$. Let (\mathcal{P}) be the set of the equations $t(q_1, \dots, q_n, q'_1, \dots, q'_r)$. We denote by P_1, \dots, P_{n+r} these equations of (\mathcal{P}) . (\mathcal{P}) will be the public key (i.e. we have “mixed” Oil and Vinegar quadratic equations and truly random quadratic equations with a secret affine permutation (\mathcal{P})).

Let $y \in K^{n+r}$ be the hash of a message M to be signed (or $y = \text{Hash}(M || 0010 || R)$) where R is a random value with no 0010 in base 2). Let $x \in K^{n+v}$. Then x is a valid signature of y if $P(x) = y$ (i.e. if $\forall i, 1 \leq i \leq n+r, P_i(x) = y_i$).

When the secret affine functions s and t are known, it is feasible to compute a valid signature after approximately $\mathcal{O}(q^r)$ computations because we will easily compute a solution for the n equations (\mathcal{A}) as before, and the probability that this solution also satisfies the r equations (\mathcal{A}') is $\frac{1}{q^r}$ (we will try again with another random R until we succeed). When q^r is small (for example if $q \leq 256$ and $r \leq 2$), this is clearly feasible. (The name “salt” comes from the fact that we cannot put a lot of salt equations since q^r must stay small for efficiency.)

Cryptanalysis when $v = n$

Here we assume that $v = n$.

Let G_i and G_j be random linear sums of the $n+r$ equations (\mathcal{P}) . The probability that G_i and G_j are linear sums of only the n equations (\mathcal{A}) is $(1/q^r)^2$ (because it is $1/q^r$ for G_i and $1/q^r$ for G_j). If this occurs, then from G_i and G_j , we will attack the scheme exactly as described in [10]. Therefore, if $v = n$, the scheme can be attacked with a complexity approximately q^{2r} (and for the legitimate user, computing a signature has a complexity approximately $\mathcal{O}(q^r)$). As a result, we do not recommend to use this variation when $v = n$.

The case $v > n$

For well chosen parameters, we have seen that we do not know how to attack Unbalanced Oil and Vinegar schemes. Therefore, of course, we do not know either how to attack the schemes when the two ideas – $v > n$ and mixing the equations with truly random equations – are combined together. However, the idea of choosing $v > n$ seems at the present to be a stronger idea (both for security and for practical implementations) than the idea of mixing Oil and Vinegar with truly random equations.

12 Another scheme: HFEV

The Unbalanced Oil and Vinegar schemes and the HFE schemes of [14] can very easily be combined, as we will see in this section. Moreover, the combined scheme looks very efficient since (at the present) we are able to avoid all the known attacks with more efficient choices of the parameters. So this HFEV schemes look both more efficient (because a smaller degree d looks sufficient for security) and more secure compared to the original HFE scheme. HFEV is also more efficient (but more complex) compared to UOV, because very few vinegar variables are needed.

The scheme (HFEV)

In the “most simple” HFE scheme (we use the notations of [14]), we have $b = f(a)$, where:

$$f(a) = \sum_{i,j} \beta_{ij} a^{q^{\theta_{ij}} + q^{\varphi_{ij}}} + \sum_i \alpha_i a^{q^{\xi_i}} + \mu_0, \quad (1)$$

where β_{ij} , α_i and μ_0 are elements of the field \mathbf{F}_{q^n} .

Let v be an integer (v will be the number of extra x_i variables, or the number of “vinegar” variables that we will add in the scheme).

Let $a' = (a'_1, \dots, a'_v)$ be a v -uple of variables of K . Let now each α_i of (1) be an element of \mathbf{F}_{q^n} such that each of the n components of α_i in a basis is a secret random linear function of the vinegar variables a'_1, \dots, a'_v .

And in (1), let now μ_0 be an element of \mathbf{F}_{q^n} such that each one of the n components of μ_0 in a basis is a secret random quadratic function of the variables a'_1, \dots, a'_v .

Then, the $n + v$ variables $a_1, \dots, a_n, a'_1, \dots, a'_v$ will be mixed in the secret affine bijection s in order to obtain the variables x_1, \dots, x_{n+v} .

And, as before, $t(b_1, \dots, b_n) = (y_1, \dots, y_n)$, where t is a secret affine bijection.

Then the public key is given as the n equations $y_i = P_i(x_1, \dots, x_{n+v})$.

To compute a signature, the vinegar values a'_1, \dots, a'_v will simply be chosen at random. Then, the values μ_0 and α_i will be computed. Then, the monovariate equations (1) will be solved (in a) in \mathbf{F}_{q^n} .

Simulations

Nicolas Courtois did some simulations on HFEV and, in all his simulations, when the number of vinegar variables is ≥ 3 , there is no affine multiple equations of small degree (which is very nice).

Example: Let $K = \mathbf{F}_2$. In HFEV, we can, for example, choose the hidden polynomial to be:

$$f(a) = a^{17} + \beta_{16} a^{16} + a^{12} + a^{10} + a^9 + \beta_8 a^8 + a^6 + a^5 + \beta_4 a^4 + a^3 + \beta_2 a^2 + \beta_1 a + \beta_0,$$

where:

- $a = (a_1, \dots, a_n)$, where a_1, \dots, a_n are the “oil” variables.
- $\beta_1, \beta_2, \beta_4, \beta_8$ and β_{16} are given by n secret linear functions on the v vinegar variables.
- β_0 is given by n secret quadratic functions on the v vinegar variables.

In this example, we compute a signature as follows: the vinegar variables are chosen at random and the resulting equation of degree 17 is solved in a .

Note: Unlike UOV, in HFEV we have terms in oil \times oil (such as a^{17} , a^{12} , a^{10} , etc), oil \times vinegar (such as $\beta_{16} a^{16}$, $\beta_8 a^8$, etc) and vinegar \times vinegar (in β_0).

13 Summary of the results for UOV

The underlying field is $K = \mathbf{F}_q$ with $q = p^m$. Its characteristic is p .

“As difficult as random” means that the problem of breaking the scheme is expected to be as difficult as the problem of solving a system of equations in v variables when the coefficients are randomly chosen (*i.e.* with no trapdoor).

Degree	Broken	Not Broken	Not broken and as difficult as random	Broken (despite as difficult as random)
2 (for all p)	$v \leq n$ or $v \simeq n$	$2n \leq v \leq \frac{n^2}{2}$	$\frac{n^2}{2} \leq v \leq n^2$	$v \geq n^2$
3 (for $p = 2$)	$v \leq (1 + \sqrt{3})n$	$(1 + \sqrt{3})n \leq v \leq \frac{n^3}{6}$	$\frac{n^3}{6} \leq v \leq \frac{n^3}{2}$	$v \geq \frac{n^3}{6}$
3 (for $p \neq 2$)	$v \leq n$ or $v \simeq n$	$2n \leq v \leq \frac{n^3}{6}$	$\frac{n^3}{6} \leq v \leq n^4$	$v \geq n^4$

In this table, we have summarized our current results on the attacks on Unbalanced Oil and Vinegar schemes. The original paper ([10]) was only studying the case $v = n$ for quadratic equations.

14 Concrete examples of parameters for UOV

In all the examples below, we do not know how to break the scheme. We have arbitrary chosen $v = 2n$ (or $v = 3n$) in all these examples (since $v \leq n$ and $v \geq n^2$ are insecure).

Example 1: $K = \mathbf{F}_2$, $n = 128$, $v = 256$ (or $v = 384$). The signature scheme is the one of section 2. The length of the public key is approximately $n \cdot (\frac{(n+v)^2}{2})$ bits. This gives here a huge value: approximately 1.1 Mbytes (or 2 Mbytes) ! The length of the secret key (the s matrix) is approximately $(n + v)^2$ bits, *i.e.* approximately 18 Kbytes. However, this secret key can always be generated from a small secret seed of, say, 64 bits.

Example 2: $K = \mathbf{F}_2$, $n = 64$, $v = 128$ (or $v = 192$). The signature scheme is the one section 8. The length of the public key is 144 Kbytes (or 256 Kbytes).

Example 3: $K = \mathbf{F}_{16}$, $n = 16$, $v = 32$ (or $v = 48$). s is a secret affine bijection of \mathbf{F}_{16} . The signature scheme is the one section 8. The length of the public key is 9 Kbytes (or 16 Kbytes).

Example 4: $K = \mathbf{F}_{16}$, $n = 16$, $v = 32$ (or $v = 48$). s is a secret affine bijection of \mathbf{F}_{16} such that all its coefficients lie in \mathbf{F}_2 . Moreover, the secret quadratic coefficients are also chosen in \mathbf{F}_2 , so that the public functions P_i , $1 \leq i \leq n$, are n quadratic equations in $(n + v)$ unknowns of \mathbf{F}_{16} , with coefficients in \mathbf{F}_2 . In this case (the signature scheme is still the one of section 8), the length of the public key is 2.2 Kbytes (or 4 Kbytes).

Note: In all these examples, $n \geq 16$ in order to avoid Gröbner bases algorithms to find a solution x (cf [6]), and $q^n \geq 2^{64}$ in order to avoid exhaustive search on x .

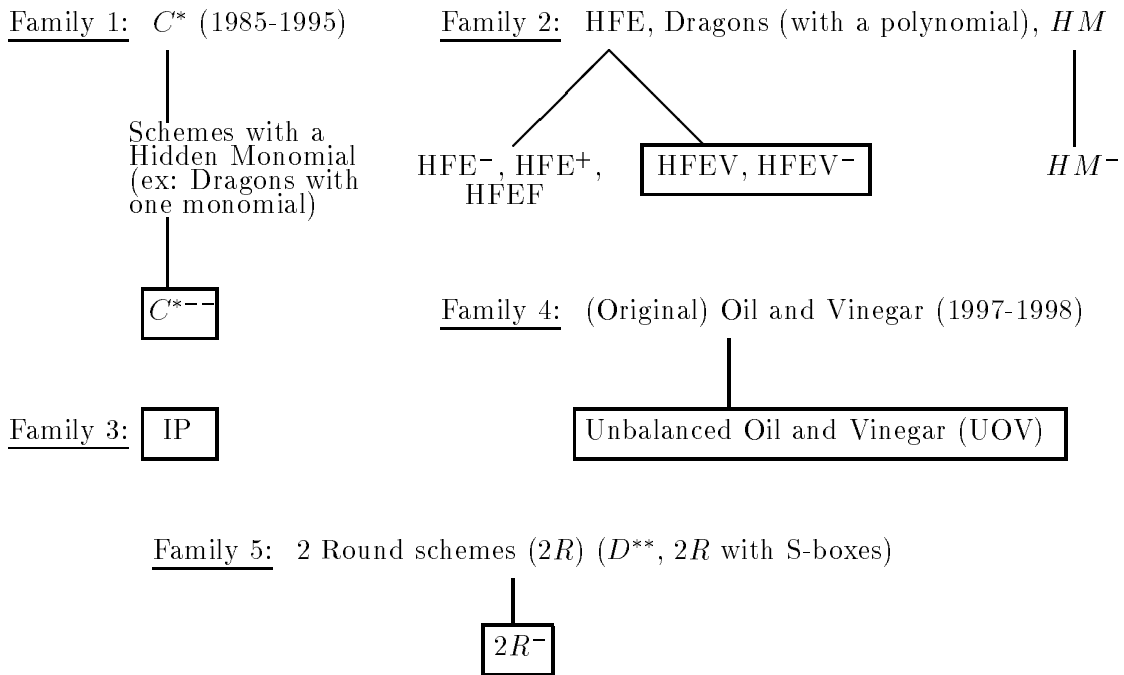
15 Concrete example of parameters for HFEV

At the present, it seems possible to choose a small value for v (for example $v = 3$) and a small value for d (for example $d = 17$ if $K = \mathbf{F}_2$).

16 State of the art (in May 1999) on Public-Key schemes with Multivariate Polynomials over a small finite field

Recently, many new ideas have been introduced to improve the schemes, such as UOV or HFEV described in this paper. Another idea is to fix some variables to hide some algebraic properties (see below). However, many new ideas have also been introduced to design better attacks on previous

schemes, such as the $-$ not yet published $-$ papers [1], [2], [3], [5]. So the field is fast moving and it can look a bit confusing at first. Moreover, some authors use the word “cryptanalysis” for “breaking” and some authors use this word with the meaning “an analysis about the security” that does not necessary mean “breaking”. In this section, we describe what we know at the present about the main schemes. In the large families of the schemes with a public key based on multivariate polynomials over a small finite field, we can distinguish between 5 main families characterized by the way the trapdoor is introduced or on the difficult problem on which the security relies. In the first family are the schemes “with a Hidden Monomial”, *i.e.* the key idea is to compute an exponentiation $x \mapsto x^d$ in a finite field for secret key computation and to “hide” such a function in the public key. In the second family are the schemes where a polynomial function (with more than one monomial) is hidden. In the third family, the security relies on an isomorphism problem. In the fourth family, the security relies on the difficulty of finding the decomposition of two multivariate quadratic polynomials from all or part of their composition. Finally, in the fifth family, the secret key computations are based on Gaussian computations. The main schemes in these families are described in the figure below. The most interesting schemes in each family are in a rectangle.



- C^* was the first scheme of all, and it can be seen as the ancestor of all these schemes. It was designed in [12] and broken in [13].
- Schemes with a Hidden Monomial (such as some Dragon schemes) were studied in [15], where it is shown that most of the simplest variations of C^* are insecure. However, C^{*--} (studied in [20]) is (at the present) the most efficient signature scheme (in time and RAM) in a smartcard. The scheme is not broken (but it may seem too simple or too close to C^* to have a large confidence in its security ...).
- HFE was designed in [14]. The most recent results about its security are in [1] and [2]. In these papers, very clever attacks are described. However, at the present, it seems that the scheme is not broken since for well chosen and still reasonable parameters the computations required to break it are still too large (moreover, asymptotically, the cryptanalysis is not polynomial if d increases as $d = \mathcal{O}(n)$ for example). For example, the first challenge of US \$500 given in the extended version of [14] has not been claimed yet (it is a pure HFE with $n = 80$ and $d = 96$ over \mathbf{F}_2).
- HFE^- is just an HFE where k of the originally public equations are not publish. Due to [1] and [2], it may be recommended to do this (despite the fact that original HFE may be secure without

it). In the extended version of [14] a second challenge of US \$500 is described on a HFE⁻. In an encryption scheme, k must be small, but in a signature scheme, k may be large.

- HFEV is described in this paper. HFEV and HFEV⁻ look very hard to break. Moreover, HFEV is more efficient than the original HFE and it can give public key signatures of only 80 bits ! In a signature scheme, the number v of “vinegar variables” can be large, but in an encryption scheme, v must be small.
- HFE⁺ is just an HFE scheme where the n originally public equations have been linearly mixed with k truly random equations. In a signature scheme, k must be small, but in an encryption scheme, k may be large.
- HFEF is just an HFE scheme where k of the variables x_i have been fixed. In a signature scheme, k must be small, but in an encryption scheme, k may be large.
- HFEVF^{+−} is just an HFE scheme where all these “perturbations” (V, F, +, −) have been applied on the public key.
- HM and HM^- were designed in [20]. Very few analysis have been done in these schemes (but maybe we can recommend to use HM^- instead of HM ?).
- IP was designed in [14]. IP schemes have the best proofs of security so far (see [19]). IP is very simple and can be seen as a nice generalization of Graph Isomorphism.
- Oil and Vinegar was presented in [16] and broken in [10].
- UOV is described in this paper. With IP, they are certainly the most simple schemes.
- $2R$ was designed in [17] and [18]. Due to [3], it is necessary to have at least 128 bits in input in the “ $2R$ with S-boxes” scheme, and due to [5], it may be wise to not publish all the (originally) public equations in all the $2R$ schemes: this gives the $2R^-$ algorithms (the efficiency of the decomposition algorithms given in [5] on the $2R$ schemes is not yet completely clear).

Remark 1: When a new scheme is found in these families, we do not necessary have to explain how the trapdoor has been introduced. Then we have a “Secret-Public Key scheme” ! The scheme is clearly a Public Key scheme since anybody can verify a signature from the public key (or can encrypt from the public key) and the scheme is secret since the way to compute the secret key computations (*i.e.* the way the trapdoor has been introduced) has not been revealed. For example, we could have done this for HFEV (instead of publishing it).

Remark 2: These schemes are of theoretical interest but (at the exception of IP) their security is not directly relied to a clearly defined and considered to be difficult problem. So is it reasonable to implement them in real products ? We think indeed that it is a bit risky to rely all the security of sensitive applications on such scheme. However, at the present, most of the smartcard applications use secret key algorithms because RSA smartcards are more expensive. So it can be reasonable to put in a low-cost smartcard one of the previous public key schemes in addition to (not instead of) the presnet secret key schemes (such as Triple-DES). Then the security can only be increased, the price of the smartcard would still be low (no coprocessor needed). The security would then rely on a master secret key for the secret key algorithm (with the risk of depending on a master secret key) and on a new low-cost public-key scheme (with the risk that the scheme has no proof of security).

17 Conclusion

In this paper, we have presented two new public key schemes: UOV and HFEV. The study of such schemes has led us to analyze very general properties about the solutions of systems of general quadratic forms. Moreover, from the general view presented in section 15, we see that these two schemes are at the present among the most interesting schemes in two of the five main families of schemes based on multivariate polynomials over a small finite field. Will this still be true in a few years ?

References

- [1] Anonymous, *Cryptanalysis of the HFE Public Key Cryptosystem*, not yet published.
- [2] Anonymous, *Practical cryptanalysis of Hidden Field Equations (HFE)*, not yet published.
- [3] Anonymous, *Cryptanalysis of Patarin's 2-Round Public Key System with S Boxes*, not yet published.
- [4] D. Coppersmith, *personal communication*, e-mail.
- [5] Z. Dai, D. Ye, K.-Y. Lam, *Factoring-attacks on Asymmetric Cryptography Based on Mapping-compositions*, not yet published.
- [6] J.-C. Faugere, *personal communication*.
- [7] H. Fell, W. Diffie, *Analysis of a public key approach based on polynomial substitutions*, Proceedings of CRYPTO'85, Springer-Verlag, vol. 218, pp. 340-349
- [8] M. Garey, D. Johnson, *Computers and Intractability, a Guide to the Theory of NP-Completeness*, Freeman, p. 251.
- [9] H. Imai, T. Matsumoto, *Algebraic Methods for Constructing Asymmetric Cryptosystems*, Algebraic Algorithms and Error Correcting Codes (AAECC-3), Grenoble, 1985, Springer-Verlag, LNCS n°229.
- [10] A. Kipnis, A. Shamir, *Cryptanalysis of the Oil and Vinegar Signature Scheme*, Proceedings of CRYPTO'98, Springer, LNCS n°1462, pp. 257-266.
- [11] R. Lidl, H. Niederreiter, *Finite Fields*, Encyclopedia of Mathematics and its applications, volume 20, Cambridge University Press.
- [12] T. Matsumoto, H. Imai, *Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption*, Proceedings of EUROCRYPT'88, Springer-Verlag, pp. 419-453.
- [13] Jacques Patarin, *Cryptanalysis of the Matsumoto and Imai public Key Scheme of Eurocrypt'88*, Proceedings of CRYPTO'95, Springer-Verlag, pp. 248-261.
- [14] J. Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP) : Two New Families of Asymmetric Algorithms*, Proceedings of EUROCRYPT'96, Springer, pp. 33-48.
- [15] Jacques Patarin, *Asymmetric Cryptography with a Hidden Monomial*, Proceedings of CRYPTO'96, Springer, pp. 45-60.
- [16] J. Patarin, *The Oil and Vinegar Signature Scheme*, presented at the Dagstuhl Workshop on Cryptography, september 1997 (transparencies).
- [17] J. Patarin, L. Goubin, *Trapdoor One-way Permutations and Multivariate Polynomials*, Proceedings of ICICS'97, Springer, LNCS n°1334, pp. 356-368.
- [18] J. Patarin, L. Goubin, *Asymmetric Cryptography with S-Boxes*, Proceedings of ICICS'97, Springer, LNCS n°1334, pp. 369-380.
- [19] J. Patarin, L. Goubin, N. Courtois, *Improved Algorithms for Isomorphisms of Polynomials*, Proceedings of EUROCRYPT'98, Springer, pp. 184-200.
- [20] J. Patarin, L. Goubin, N. Courtois, C_{-+}^* and HM: Variations Around Two Schemes of T. Matsumoto and H. Imai, Proceedings of ASIACRYPT'98, Springer, pp. 35-49.
- [21] A. Shamir, *A simple scheme for encryption and its cryptanalysis found by D. Coppersmith and J. Stern*, presented at the Luminy workshop on cryptography, september 1995.