CRYPTANALYSIS OF $\mathcal{HFE}$

ILIA TOLI

Abstract. Out of the public key ($\mathcal{PK}$) we recover a polynomial of the same shape as the private polynomial. Then we give an algorithm for solving such a special-form polynomial. This fact puts an eavesdropper in the same position with a legitimate user in decryption. An upper bound for the complexity of that all is $\mathcal{O}(n^6)$ bit operations for n the degree of the field extension.

1. Introduction

The problem of solving systems of multivariate polynomial equations is a well-known hard problem. In complexity theory, it is well-known to be an $\mathcal{NP}$-complete problem. Furthermore, even if we limit ourselves to the problem of solving systems of multivariate polynomial of degree two equations, we have again an $\mathcal{NP}$-complete problem. Therefore, it has been paid a lot of attention, since the invention of the idea of the $\mathcal{PK}$ cryptography, by Diffie and Hellman [DH76].

A lot of cryptosystems have been proposed since then, where an eavesdropper is asked to accomplish the hard task of solving systems of quadratic equations. However, most of them had short lives. The information that an eavesdropper had on the shape of the private key usually sufficed to compromise the security. Some of their cryptanalyses aimed to recover the private key, or something equivalent, in the sense that gives the same privileges. Other cryptanalyses reduce the problem to accessible exhaustive searches, and so on. Recall that the ultimate task of the cryptanalysis is recovering cleartexts, and not recovering meticulously the whole set of the values of the $\mathcal{PK}$ [COU].

In this paper we focus on $\mathcal{HFE}$. It is a $\mathcal{PK}$ cryptosystem first proposed by Patarin [Pat96]. It is one of the modifications of a cryptosystem first proposed by Imai and Matsumoto [IM85], after having successfully cryptanalyzed it.

In its main version, its $\mathcal{PK}$ is a system of n quadratic polynomial equations in n variables with coefficients in a finite field $\mathbb{F}_q$, practically $\mathbb{F}_2$. Its private key is:

- a basis, up to an isomorphism, of an overfield $\mathbb{K} \supset \mathbb{F}_q$, $[\mathbb{K} : \mathbb{F}_q] = n$, as an $\mathbb{F}_q$-vector space;

1991 *Mathematics Subject Classification.* Primary: 11T71; Secondary: 12H05.

Key words and phrases. Public key cryptography, hidden monomial, hidden field equations ($\mathcal{HFE}$), polynomial system solving.

- a univariate polynomial f of a certain form, with coefficients in $\mathbb{K}$;
- two nondegenerate affine transformations of $\mathbb{K}$.

Practically, $p = q = 2$. However, for simplicity, hereon we assume only that $p = q$. The other case can be treated identically.

In the our cryptanalysis, we find another sparse univariate polynomial, such that its knowledge reduces eavesdropping to the task of solving a single univariate polynomial equation. We call it *an alias of the* $\mathcal{PK}$. All of this task can be performed within $\mathcal{O}(n^6)$ bit operations. Recall that n is actually the only security parameter to the legitimate user, and that the trapdoor problem is subexponential in it.

We assume that the reader is already familiar with $\mathcal{HFE}$.

Most of the symbolic manipulations throughout this paper are done by means of Singular, Macaulay2, and CoCoA. If there ever are any calculus mistakes, it is because of the little part done by hand. In any case, the calculus errors in the examples do not prejudice the algorithms they illustrate.

2. The Cryptosystem

Let the parties committed to the tasks be:

- Alice who wants to receive secure messages;
- Bob who wants to send her secure messages;
- Eve, the eavesdropper.

Alice chooses two finite fields $\mathbb{F}_q < \mathbb{K}$, and a basis $\beta_1, \beta_2, \ldots, \beta_n$ of $\mathbb{K}$ as an $\mathbb{F}_q$-vector space. In practice, $q = 2$. However, it can be any p^r, for any p prime, and any $r \in \mathbb{N}$.

Next she takes a univariate polynomial of the form:

$$f(x) = \sum_{i,j} \gamma_{ij} x^{q^{\theta_{ij}} + q^{\varphi_{ij}}} + \sum_i \alpha_i x^{q^{\xi_i}} + \mu_0, \tag{1}$$

with coefficients in $\mathbb{K}$, and two affine transformations: $\mathcal{S}, \mathcal{T} : \mathbb{K} \to \mathbb{K}$; one left, one right. Let ∂_f be the degree (private data) of $f(x)$.

With manipulations that we skip in order to save space, she generates her $\mathcal{PK}$; a set of n quadratic polynomials of degree two, in n variables. The interested reader can find details in [IM85, IM89, hfe, Tol03].

Her private key is:

- the basis B of $\mathbb{K}$ as an $\mathbb{F}_q$-vector space;
- $\mathcal{S}$, f, $\mathcal{T}$.

3. The Cryptanalysis

Applying affine transformations is equivalent to composing with affine polynomials. So, Eve knows that $\mathcal{S} \circ f \circ \mathcal{T}$ in $\mathbb{K}[x]$ is a certain univariate polynomial of the same form (1). Let Eve fix the canonical

basis of $\mathbb{K}$ (or a basis at her choice, too). She may assume to apply a nondegenerate linear transformation $\mathcal{L}$ (that she does not know, but she need not) to the private basis B of $\mathbb{K}$, and to $\mathcal{S} \circ f \circ \mathcal{T}$ in $\mathbb{K}[x]$. So, she obtains the canonical basis I of $\mathbb{K}$, and another polynomial $\mathcal{A} = \mathcal{S} \circ f \circ \mathcal{T} \circ \mathcal{L}$, of the same form like f in (1). So, it is rather sparse, too. This is very easily seen if one thinks of the affine and linear transformations as affine and linearized polynomials. Applying these affine or linear transformations is equivalent to composing with such polynomials. Now, if one observes the form of the polynomial compositum for such polynomials, one can easily reach to the conclusion that it too is of the form (1).

Alice has limitations on the degree f. Her decryption is exponentially harder with its growth. Besides, if it is too big, the number undesired solutions grows a lot. To discard them, she introduces other, randomly chosen polynomials. This renders the $\mathcal{PK}$ overdefined. So, the cryptosystem becomes particularly weak to certain attacks.

Eve may write down the pseudoquadratic polynomial of degree q^{n-1} in its general form:

$$A_d x^d + A_{d-1} x^{d-1} + \cdots + A_1 x + A_0, \tag{2}$$

where she considers the A_i like variables. She includes in such a polynomial only monomials which's exponents have Hamming weight at most two. So, her number of variables is at most n^2.

Next, Eve has to do at most n^2 evaluations to the $\mathcal{PK}$. So, she obtains a linear system of at most n^2 equations in the n^2 variables A_i. Solving it in $\mathbb{K}$ enables Eve to recover $\mathcal{A} = \mathcal{S} \circ f \circ \mathcal{T} \circ \mathcal{L}$ in the form of a univariate polynomial with coefficients in $\mathbb{K}$. It is a public knowledge that $\mathcal{A}(x)$ exists, and is unique. So, we expect that n^2 evaluations are necessary, and suffice.

Now Eve has reduced eavesdropping problem to the problem of solving a single univariate polynomial equation of a certain form and structure within its field of coefficients. Almost the same like Alice. Eve possesses the private key, indeed an alias of its. The only problem to Eve is that such a polynomial generally is of a huge degree. However, Eve knows that it is isomorphic to a very low degree polynomial of a certain form.

3.1. **Trying to Generate an $m \times m$ Quadratic System of Equations.** Sometimes Eve may try to generate a "$\mathcal{PK}$" out of the polynomial that she finds. That is, she may fix a subfield $\mathbb{S}$ of $\mathbb{K}$, $[\mathbb{K} : \mathbb{S}] = m$, for m a number of variables that she can handle to calculate Gröbner bases. Doing so, she can impose $x^{\frac{n}{m}} - x = 0$. Now she is assumed able to solve the path of univariate equations that come out of the calculation of a Gröbner basis, that she can, too.

Another tool that Eve may appeal to is factorization. She may substitute a given ciphertext to the polynomial she finds, and try to factor it within $\mathbb{K}$. She is not interested on the irreducible factors. She can find the cleartext among the first-degree factors.

3.2. For most of the rest of this paper we give a step by step example of how do we practically recover $\mathcal{A}(x)$, and then we focus the our attention about finding the roots of such a very special-form polynomial. The toy values of the parameters in the examples do not prejudice the generality of the algorithms.

4. A Toy Example

We are given the following toy $\mathcal{PK}$ from Wolf [Wol03]:

$$\begin{cases} x_1 + x_3 + x_1x_2 + x_1x_3 + x_2x_3 \\ x_3 + x_1x_3 + x_2x_3 \\ x_1 + x_2 + x_3 + x_1x_2 + x_2x_3 + 1. \end{cases} \tag{3}$$

All what we know besides the $\mathcal{PK}$ equations, is that the base field is $\mathbb{F}_2$, and that the degree of field extension is *3*. In some fashion, we will have these data public. Without them, Bob will be unable to encrypt.

We fix the basis $t^2, t, 1$ of $\mathbb{K} = \mathbb{F}_{2^3}$ as an $\mathbb{F}_2$-vector space. We choose it at our pleasure. We take $\mathbb{K} = \mathbb{F}_2[t]/(t^3 + t + 1)$. Again, we choose the irreducible polynomial of degree n from $\mathbb{F}_2[t]$ for generating $\mathbb{K}$ at our pleasure.

Now we write the general form of the polynomial we are looking for; an alias of the private polynomial f. It has at most $3^2 = 9$ terms.

Explicitely, in this case it is of the form:

$$a + bx + cx^2 + dx^3 + ex^4 + fx^5 + gx^6. \tag{4}$$

Now we evaluate the $\mathcal{PK}$ in *7* points: $x = 0, 1, t, t+1, t^2, t^2+1, t^2+t$.

The toy values of the parameters render the wrong idea that we will have to evaluate a generic-coefficients polynomial in the whole set of the elements of the overfield. Indeed, it is very far from being like that. We need only n^2 evaluations. Card $\mathbb{K} = p^n$, instead.

From the evaluations we obtain the following system:

$$\begin{cases} a = 1 \\ a + b + c + d + e + f + g = t^2 \\ a + tb + t^2c + (t+1)d + (t^2+t)e + (t^2+t+1)f + (t^2+1)g = 0 \\ a + (t+1)b + (t^2+1)c + t^2d + (t^2+t+1)e + tf + (t^2+1)g = 0 \\ a + t^2b + (t^2+t)c + (t^2+1)d + te + (t+1)f + (t^2+t+1)g = t^2 \\ a + (t^2+1)b + (t^2+t+1)c + (t^2+t)d + (t+1)e + t^2f + tg = t^2 + 1 \\ a + (t^2+t)b + tc + (t^2+t+1)d + t^2e + (t^2+1)f + (t+1)g = 1. \end{cases}$$

We solve this system, and find the our alias key:

$$\mathcal{A}(x) = t^2x^6+(t^2+1)x^5+(t^2+t+1)x^4+(t^2+1)x^3+(t^2+t)x^2+1. \tag{5}$$

As the polynomial we are looking for is unique, the solution to the system above exists, and is unique. Now Eve has only to solve the equation $\mathcal{A}(x) = y$ in order to recover x. Even though it is of an enormous degree, the number of solutions that Eve finds is equal to those that Alice is expected to find. This is a public knowledge. Eve, too, can descard undesired solutions by the same means that Alice does. Much the same like Alice.

Up to now we do not break $\mathcal{HFE}$. However, having this polynomial explicit, and for such a little computational effort does not help its security. At least, Alice has now to care that the polynomial that Eve finds must be hard to solve by all means. Besides, the coefficients of this polynomial enclose a lot of informations about the coefficients of the affine polynomials that were employed for its generation. Recall, eg., that during the whole set of transformations the coefficients of the quadratic monomials were not influenced by the coefficients of the linearized monomials.

For the rest, the transformations that did we apply to f in order to obtain $\mathcal{A}(x)$ may increase or decrease the degree of $\mathcal{A}(x)$, too. Besides, Eve now can try to compose her polynomial $\mathcal{A}(x)$ with affine polynomials modulo $x^{p^n} - x$, in order to obtain a lower-degree polynomial, and pass in the position of Alice, or close to it, or even to decompose it modulo $x^{p^n} - x$. This is the topic of the next section.

5. Composition and Decomposition Modulo $x^{p^n} - x$

Henceforth we get rid of the $\mathcal{PK}$. Most of the remainder of this paper deals with solving $\mathcal{A}(x)$. Henceforth we denote $\mathcal{A}(x) = a(x)$. We give next a special-purpose rootfinding algorithm of a very modest computational effort. We compose and decompose $a(x)$ in some way that makes its degree decrease.

Given a univariate polynomial, we can associate to it a Hamming weight in several fashions. The following one is particularly useful for the our purposes.

Definition 5.1. *We call* Hamming weight *of a polynomial $a(x)$ to be the maximum of the Hamming weights of the exponents of its monomials. We denote it by $\mathcal{H}(a)$.*

If one of two polynomials has it equal to one, their compositum has the Hamming weight equal to the other. The Hamming weight of a remainder modulo $(x^{p^n} - x)$ is equal to the Hamming weight of the dividend. This is all what do we need about it.

We are given a certain polynomial $a(x)$. We assume it practically infeasible for us to find its roots directly. Therefore, we want to try to find some affine polynomial $b(x)$, such that either $a \circ b$, or $b \circ a$ are of a moderate degree. Next, we may try to find a polynomial of moderate degree $c(x)$, and an affine polynomial $d(x)$ such that either $c \circ d = a$, or $d \circ c = a$.

Each of such informations that we may get, renders polynomial solving pretty easy to us.

Let us consider now the our case, ie, when $\mathcal{H}(a) = 2$. We want to find an affine polynomial $d(x)$, such that $d \circ a \mod (x^{p^n} - x)$ has a moderate degree. Let us write the equation in its general form:

$$(d \circ a)(x) = b(x) \cdot (x^{p^n} - x) + r(x). \tag{6}$$

We are interested on finding $r(x)$ and $d(x)$. The polynomial $r(x)$ has the same roots like $(d \circ a)(x)$. The polynomial $d(x)$ instead is always pretty easily invertible, in the sense that its roots can always be very easliy found, independently of its degree. It takes a linear system of equations.

So, knowing the roots of $r(x)$ we can very easily detect the roots of $a(x)$, that we are interested to solve. In its general form, the polynomial d has at most n coefficients.

On the polynomial $r(x)$. We dont know its degree. Besides, it is not unique. We know its form. It contains only monomials of Hamming weight at most two. Among all its possible choices, we are interested to take the one of lowest degree, without falling into degenerate cases. It will be the hard part of polynomial solving. Well, we have already one choice of $r(x)$. It is $a(x)$ itself. So, the polynomials we are searching for, have at most $(log_p \partial_a)^2$ terms. Clearly, we are not interested on the other choices.

Now we are ready to write the relations among the polynomials involved. We write both $d(x)$ and $r(x)$ in their general form. What dont we know are their coefficients. We consider them variables.

We calculate $(a(x))^{p^i} \mod (x^{p^n} - x)$ for $i = 0, \ldots, n-1$. We put these polynomials respectively as coefficients of $d \circ a$ in its generic form. So doing we have obtained essentially a single linear equation in $n + (log_p \partial_a)^2$ variables. Now to determine a set of suitable coefficients we have only to make at most $n + (log_p \partial_a)^2$ evaluations of the $\mathcal{PK}$.

So, we have obtained a system of $n + (log_p \partial_a)^2$ linear equations in $n + (log_p \partial_a)^2$ variables. If we only assume that $\partial_{f \circ \mathcal{T} \circ \mathcal{L}} < \partial_{\mathcal{S} \circ f \circ \mathcal{T} \circ \mathcal{L}}$, it must have a solution. Besides, it may have other solutions, however.

Next, among all the solutions that we may find, we are particularly interested about the unique one where the first nonzero coefficient of $r(x)$ has the index lowest possible (unique, if we take $r(x)$ to be monic. It is not any limitation.).

Let us now give a step by step description of how do we determine the coefficients of the polynomials we are interested for. Let us continue with the example above. Again, due to the toy values we are not able at all to make the whole set of required evaluations. However the example is a best illustrative one.

6. Coefficient Determining

It is a public knowledge that $a(x)$ is isomorphic to a low-degree polynomial. We do best finalize our work if $r(x)$ has low degree, and $d(x)$ is a permutation polynomial. If $d(x)$ is not, it helps still, in the sense that we decrypt most of the ciphertexts.

The general form of the our equation is:

$$\begin{gathered}(a+bx+cx^2+dx^4+ex^6)\circ\\ \circ(t^2x^6+(t^2+1)x^5+(t^2+t+1)x^4+(t^2+1)x^3+(t^2+t)x^2+1)=\\ =f+gx+hx^2+ix^3+jx^4+kx^5+mx^6.\end{gathered}$$

In order to increase the our chance on finding a permutation $d(x)$, we may impose $b=1$, ie, the coefficient close to the monomial x of $d(x)$. However, in general we do not make any assumption.

In its full expansion, the identity above becomes:

$$\begin{aligned}&a+b(t^2x^6+t^2x^5+t^2x^4+t^2x^3+t^2x^2+tx^4+tx^2+x^5+x^4+x^3+1)+\\&+c(t^2x^6+t^2x^5+t^2x^3+tx^6+tx^5+tx^4+tx^3+tx+x6+x^3+x+1)+\\&+d(t^2x^2+t^2x+tx^6+tx^5+tx^3+x^6+x^5+x^2+1)+\\&+e(t^2x^4+t^2x^3+t^2x+tx^7+tx^5+tx^4+tx^3+tx^2+x^7+x^2+1).\end{aligned}$$

Doing all the n^2+n evaluations we provided, and in this tiny cryptosystem all what we can, and less than provided, we obtain the following system of equations:

$$\begin{cases}a+b+c+d+e=f\\ a+bt^2+c(t^2+t)+dt+e(t^2+t+1)=f+g+h+i+j+k+m\\ a=f+gt+ht^2+i(t+1)+j(t^2+t)+k(t^2+t+1)+mt^2+m\\ a+b(t^2+t+1)+c(t+1)+d(t^2+1)+et^2=\\ =f+g(t+1)+h(t^2+1)+it^2+j(t^2+t+1)+kt+m(t^2+t)\\ a+bt^2+c(t^2+t)+dt+e(t^2+t+1)=\\ =f+gt^2+h(t^2+t)+i(t^2+1)+jt+k(t+1)+m(t^2+t+1)\\ a+b(t^2+1)+c(t^2+t+1)+d(t+1)+et=\\ =f+g(t^2+1)+h(t^2+t+1)+i(t^2+t)+j(t+1)+kt^2+mt\\ a+b+c+d+e=\\ =f+g(t^2+t)+ht+i(t^2+t+1)+jt^2+k(t^2+1)+m(t+1)\\ a+bt+ct^2+d(t^2+t)+e(t^2+1)=\\ =f+g(t^2+t+1)+h(t+1)+it+j(t^2+1)+k(t^2+t)+mt^2.\end{cases}$$

Well, here is a first facility. Given $a(x)$ and $d(x)$, the polynomial $r(x)$ is uniquely determined. So, the rank of the matrix of the system above is at most n, the number of coefficients of $d(x)$. So, we need at most n evaluations, and $2n$ variables.

What we are seeking for is a solution of its that renders zero the coefficient of m and k and j, and so on, as many as possible, and, best that may happen, that $d(x)$ is a permutation polynomial. If this is the position, we are done. However, any of the solutions of the above system helps rootfinding.

Besides, at worst, it is however simple for us to calculate one couple $(d(x), r(x))$ for each reasonable degree of $r(x)$, ie take several solutions of the above system.

6.1. We solve the system above by Gaussian elimination. We are interested to get a $r(x)$ nontrivial of smallest degree possible. Therefore, we eliminate for last the coefficients from righthand side corresponding to the highest degree monomials. Then we can decide to render zero them all but one, and go up in the Gaussian stair.

$$\begin{cases} h(t+1)+i+j+mt=0 \\ g(t^2+t)+ht+i(t^2+t+1)+jt^2+k(t^2+1)+m(t+1)=0 \\ e(t+1)+h(t^2+t+1)+i(t+1)+jt^2+k+m=0 \\ d(t^2+1)+e(t^2+1)+gt^2+h(t^2+t)+it^2+j+k(t+1)+mt=0 \\ c+d(t+1)+e(t^2+1)+gt^2+h+i(t^2+t)+k(t+1)+m(t^2+t+1)=0 \\ b+c+d+e+gt+ht^2+i(t+1)+j(t^2+t)+k(t^2+t+1)+m(t^2+1)=0 \\ a+b+c+d+e+f=0. \end{cases}$$

We have several choices now. We want to have lowest possible the degree of $r(x)$, without falling into trivial solutions. So, we want to render zero as many as possible of the variables $m > k > j > i > h > g > f$, in order. What can we do is to take few of them, then choose. The first solution is: we render zero all of the variables in the first equation. Then the system becomes:

$$\begin{cases} h=i=j=m=0 \\ g(t^2+t)+k(t^2+1)=0 \\ e(t+1)+k=0 \\ d(t^2+1)+e(t^2+1)+gt^2+k(t+1)=0 \\ c+d(t+1)+e(t^2+1)+gt^2+k(t+1)=0 \\ b+c+d+e+gt+k(t^2+t+1)=0 \\ a+b+c+d+e+f=0. \end{cases} \tag{7}$$

Now, if we choose $k = 0$, we fall into the trivial case. So, we choose $k \neq 0$. Eg, $k = t$. By this case we have:

$$\begin{cases} h = i = j = m = 0 \\ k = t, \quad g = t + 1, \quad e = t^2 + t \\ d = t^2, \quad c = t^2 + 1, \quad b = t^2 \\ a + f = t + 1. \end{cases} \tag{8}$$

So, one choice should be:

$$r_1(x) = tx^5 + (t+1)x + 1, \quad d_1(x) = (t^2+t)x^6 + t^2x^4 + (t^2+1)x^2 + t^2x + t.$$

If we take $m = k = j = 0$, we find one solution to be:

$$\begin{cases} m = k = j = 0, \quad i = 1, \quad h = t^2 + t \\ g = 0, \quad d = t^2, \quad c = 0, \quad b = t^2 + 1 \\ a + f = t + 1. \end{cases} \tag{9}$$

Now we can choose:

$$r_2(x) = x^3 + (t^2 + t)x^2 + 1, \quad d_2(x) = (t + 1)x^6 + t^2x^4 + (t^2 + 1)x + t.$$

This is one of the best pairs we can hope for, as *3*, the degree of $r(x)$, is the smallest number of Hamming weight two. Unfortunately, $d_2(x)$ is not a permutation. However, with better choices of coefficients we can obtain this condition, too.

6.2. Once we have a suitable $d(x)$, $r(x)$, it is pretty simple now to solve $a(x)$. Explicitely, if $d(x)$ is a permutation polynomial, we find all the roots r_i of $r(x)$, then calculate $d(r_i)$. So we obtain all the roots of $a(x)$. If $d(x)$ is not a permutation polynomial, anyway we can calculate most of the solutions of $a(x)$.

Example 6.1. *Encrypting $t^2 + 1$, we obtain $\mathcal{A}(t^2 + 1) = t^2 + 1$. Eve knows t^2+1, this ciphertext. She solves $r_1(x) = t^2+1$, and finds $x = t$. She calculates $d_1(t) = t^2 + 1$, and has succeeded decryption.*

Example 6.2. *Encrypting $t^2 + t + 1$, we have $\mathcal{A}(t^2 + t + 1) = t$. Eve solves $r_2(x) = t$, and finds $x = t^2+t+1$. She calculates $d_2(t^2+t+1) = 1$. She did not succeed decryption. Surprizingly enough, the cleartext is the root of the first equation.*

7. The Rank

The final tile we miss in order to complete the cryptanalysis of $\mathcal{HFE}$ is to prove that by the algorithm described in the previous section we can always obtain good polynomials $d(x)$ and $r(x)$. Ie, that we can find $r(x)$ with degree of the same order of magnitude like $f(x)$, in order that then we are able to solve it directly, like Alice does. For that, it

suffices that we prove that the rank of the matrix of the homogeneous linear system of equations we are required to solve does not exceed the number of nonzero coefficients of the private polynomial.

We omit the other problem of existence of a permutation $d(x)$. It does always exist. However, even if we do not make such an assumption, the security of $\mathcal{HFE}$ is already prejudiced by the nonpermutation polynomials $d(x)$.

We can now get rid of the transformation $\mathcal{S}$. If the $\mathcal{PK}$ is of the form $\mathcal{S} \circ \mathcal{P}$ for some $\mathcal{P}$, we know how to factor out $\mathcal{S}$, or $\mathcal{S}$ and some other factor of $\mathcal{P}$. Ie, we know how to obtain $r(x)$ of degree at most $\partial_{\mathcal{P}}$.

Being given $\mathcal{A}$, the polynomial $\mathcal{T} \circ \mathcal{L}$ is uniquely determined by f. So, there suffice to vary its coefficients in order to obtain the whole set of matrices we use in order to factor $\mathcal{A}$ in the left side. So the rank of any such a matrix does not exceed the number of the coefficients of f.

8. Conclusions

8.1. In $\mathcal{HFE}$ the $\mathcal{PK}$ hides a single univariate pseudoquadratic polynomial. In any fashion, this polynomial is very sparse. It has no more than n^2 terms of a certain well-known shape. So, in any case, Eve can recover it in $\mathcal{O}(n^6)$ bit operations, for n the degree of the field extension. Recall that n is Alice's only security parameter, and that the trapdoor problem is already only subexponentially harder with it.

8.2. Even if we take the private polynomial to be of higher Hamming weight, the amount of calculi required to recover it is almost the same. Recall that the size of the $\mathcal{PK}$ is already almost impractical.

8.3. Solving univariate polynomial equations upon finite fields is a time-honoured hard problem. So, it is reasonable to look for cryptosystems that provide it as a trapdoor problem. The experience up to now has shown that hiding polynomials does not help the security of a cryptosystem, restricts choices, and renders the size of the $\mathcal{PK}$ impractical. The privileged position of a legitimate user must rely elsewhere.

9. Cryptosystems that Might be Secure

9.1. Alice takes a finite field $\mathbb{F}_{p^n}$. Next she chooses k polynomials: $f_1, \ldots, f_k$ from $\mathbb{F}_{p^n}[x]$, of any degree. Then she calculates:

$$f_1 \circ f_2 \cdots \circ f_k = a(x) \cdot (x^{p^n} - x) + r(x). \tag{10}$$

The $\circ$ stand for functional composition of polynomials. We assume that the remainder is proper, i.e., $a(x) \neq 0$. If we did not provide reductions modulo $(x^{p^n} - x)$, the complexity of that all is $\mathcal{O}((log\ p) \cdot n^3)$. The algorithm is refferred to as *square and multiply*, or *repeated square*. Provided the reductions modulo $(x^{p^n} - x)$, it all is far easier.

Alice publishes $r(x)$. It is her $\mathcal{PK}$, together with the field $\mathbb{F}_{p^n}$, and an alphabet. We assume $r(x)$ is a polynomial of an enormous degree. Best if it is as high as $\approx p^n$. An eavesdropper is assumed unable to solve it directly. We assume that $r(x)$ has no structure that renders it easy to solve directly. That is, it may be nondecomposable, etc.

Bob to encrypt a message $m \in \mathbb{F}_{p^n}$, evaluates the public polynomial in this point, and sends it to Alice.

Alice to decrypt finds all different roots of $r(x) - r(m)$ within $\mathbb{F}_{p^n}$. She can, as she can convert the task in finding roots of each composand. As $m^{p^n} - m = 0$, it is easily seen that if $m \in \mathbb{F}_{p^n}$ is a cleartext, we have:

$$f_1 \circ f_2 \cdots \circ f_k(m) = r(m). \tag{11}$$

Next, Alice has to limit the number of total solutions, and to distinguish the right solution among the others. This is a technical problem. There exist already a lot of instruments to handle it.

A good way to limit the undesired solutions is to take all of the f_i to be permutation polynomials, but f_1. The polynomial f_1 is taken to be of a very moderate degree. deg f_1 is a bound on the number of the solutions.

We may be attempted to take all of the f_i to be permutation polynomials. By this case, $r(x)$, the $\mathcal{PK}$, is a permutation polynomial, too. So, decryption is deterministic. However, such polynomials have shown to be particularly bad in cryptography.

Some of the polynomials employed in key generation may be permutation polynomials, affine polynomials of a huge degree, and so on. I mean that they can be chosen in order to render the task easier to Alice. In any case, k will be a very small natural number. The affine polynomials have the very nice property that if suitably chosen, they have huge multiplicities. The suitable choices are a plethora.

9.2. A major variation of these settings may consist on generating the $\mathcal{PK}$ to be a "randomly chosen" bivariate polynomial, again of very high in x, and of a moderate degree in y. Given the moderate degree in y, Bob is assumed to be able to solve it with respect to y, in order to find a ciphertext.

9.3. Both Alice and Eve are interested on solving the public polynomial within $\mathbb{F}_{p^n}$. That's why we reduce modulo $(x^{p^n} - x)$. If we take the public polynomial bivariate, we reduce modulo $(y^{p^n} - y)$, too.

In other variations, we may take the ciphertext to be longer than the cleartext, i.e., rely in an overfield of $\mathbb{F}_{p^n}$, and so on.

9.4. Throughout, the only security parameter is the degree of the public polynomial in x. The trapdoor is assumed to be exponentially harder with it. The bigger n, the exponentially bigger can be taken the

degree of the public polynomial. So, the trapdoor problem is assumed to be twice exponentially hard with n.

9.5. A beautiful variation should be the following. Alice assumes there will be a handful of very few impossible cleartexts $m_1, m_2, \ldots, m_\ell \in \mathbb{F}_{p^n}$. She assumes herself unable to decrypt them, and assumes that they will never be sent. Next she calculates:

$$t(x) = (x-m_1)\cdot(x-m_2)\cdots(x-m_\ell), \quad \text{then} \quad w(x) = \frac{x^{p^n} - x}{t(x)}. \tag{12}$$

Now she builds her $\mathcal{PK}$ to be $r(x)$ as in:

$$f_1 \circ f_2 \cdots \circ f_k = a(x) \cdot w(x) + r(x). \tag{13}$$

Now everything that Eve knows is that the $\mathcal{PK}$ is the remainder of some polynomial modulo some polynomial. It does not seem very much for her to start a cryptanalysis. It seems to be a "zero-knowledge $\mathcal{PK}$".

9.6. Yet a lot of other variations are of course possible. Alice may, e.g., take $w(x)$ as above. Next she takes a random polynomial $g(x) \in \mathbb{F}_{p^n}[x]$, and calculates:

$$f_1 \circ f_2 \cdots \circ f_k = a(x) \cdot g(x) \cdot w(x) + r_1(x), \tag{14}$$

and then:

$$r_1(x) = b(x) \cdot (x^{p^n} - x) + r(x). \tag{15}$$

The polynomial $r(x)$ is her $\mathcal{PK}$.

9.7. One of the attentions of Alice is that the Hamming weight of her $\mathcal{PK}$ polynomial is huge. So, even when Eve succeeds to factor it somehow, she is still required to solve a huge-degree polynomial.

References

[COU] http://www.cryptosystem.net/ttm/.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. In *IEEE Trans. Information Theory*, pages 644–654, 1976. http://www.cs.jhu.edu/~rubin/courses/sp03/papers/diffie.hellman.pdf.

[hfe] http://www.minrank.org/hfe/ or http://www.hfe.info/.

[IM85] Hideki Imai and Tatsuo Matsumoto. Algebraic methods for constructing asymmetric cryptosystems. In *Algebraic Algorithms and Error-Correcting Codes, Proceedings Third International Conference*, pages 108–119, Grenoble, France, 1985. Springer-Verlag.

[IM89] Hideki Imai and Tatsuo Matsumoto. Public quadratic polynomial tuples for efficient signature-verification and message-encryption. In *Advances in Cryptology, Eurocrypt '88*, pages 419–453. Springer-Verlag, 1989. http://link.springer.de/link/service/series/0558/papers/0330/03300419.pdf.

[Pat96] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. *Lecture Notes in Computer Science*, 1070:33–on, 1996. http://www.minrank.org/hfe.pdf.

[Tol03] Ilia Toli. Hidden polynomial cryptosystems. Cryptology ePrint Archive, Report 2003/061, 2003. http://eprint.iacr.org/2003/061.pdf.

[Wol03] Christopher Wolf. Efficient public key generation for multivariate cryptosystems. Cryptology ePrint Archive, Report 2003/089, 2003. http://eprint.iacr.org/.

Dipartimento di Matematica *Leonida Tonelli*, via F. Buonarroti 2, 56127 Pisa, Italy., toli@posso.dm.unipi.it