Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms

Jacques PATARIN

CP8 Transac,
68 route de Versailles, BP 45,
78431 Louveciennes Cedex, France

Abstract. In [6] T. Matsumoto and H. Imai described a new asymmetric algorithm based on multivariate polynomials of degree two over a finite field, which was subsequently broken in [9]. Here we present two new families of Asymmetric Algorithms that so far have resisted all attacks, if properly used: Hidden Field Equations (HFE) and Isomorphism of Polynomials (IP). These algorithms can be seen as two candidate ways to repair the Matsumoto-Imai Algorithm. HFE can be used to do signatures, encryption or authentication in an asymmetric way, with very short signatures and short encryptions of short messages. IP can be used for signatures and for zero knowledge authentication.
An extended version of this paper can be obtained from the author. Another way to repair the Matsumoto-Imai Algorithm will be presented in [10].

1 Introduction

Currently the security of most algorithms that we know in Asymmetric Cryptography for encryption or signatures relies on the unproved intractability of the integer factorization or discrete log problem. One of the challenges of Asymmetric Cryptography is to find new and efficient algorithms for encryption or signatures that do not depend on these two closely related problems. For authentication the situation is much better, due to the algorithms presented for example in [12] and [13].

In this paper we propose two new classes of Asymmetric Algorithms whose security does not depend on factoring or discrete logs: Hidden Field Equations (HFE) and Isomorphism of Polynomials (IP). Furthermore HFE also address two other problems facing Asymmetric Cryptography: generating very short asymmetric signatures, and generating short encryptions of short messages. Both HFE and IP are based on a scheme described by T. Matsumoto and H. Imai (cf. [6]). In [9] an efficient attack against this scheme was presented. Unfortunately, we are not able to prove the security of HFE or IP either, but so far they have resisted all attacks, including the one from [9]. Moreover IP-based authentications can be proved to be zero-knowledge.

U. Maurer (Ed.): Advances in Cryptology - EUROCRYPT '96, LNCS 1070, pp. 33-48, 1996.

2 Preliminaries

Throughout this paper we use the following notation. We denote by $\mathbf{F}_q$ a finite field of cardinality q and characteristic p, for some prime p and prime power $q = p^m$. let $\mathbf{F}_{q^n}$ be an extension of degree n of $\mathbf{F}_q$. Let

$$f(x) = \sum_{i,j} \beta_{ij} x^{q^{\theta_{ij}}+q^{\varphi_{ij}}} + \sum_{k} \alpha_k x^{q^{\xi_k}} + \mu \in \mathbf{F}_{q^n}[x]$$

be a polynomial in x over $\mathbf{F}_{q^n}$ of degree d, for integers $\theta_{ij}, \varphi_{ij}, \xi_k \geq 0$.

Since $\mathbf{F}_{q^n}$ is isomorphic to $\mathbf{F}_q[x]/(g(x))$, if $g(x) \in \mathbf{F}_q[x]$ is irreducible of degree n, elements of $\mathbf{F}_{q^n}$ may be represented as n-tuples over $\mathbf{F}_q$, and f may be represented as a polynomial in n variables x_1, x_2, ..., x_n over $\mathbf{F}_q$:

$$f(x_1, \ldots, x_n) = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, x_2, \ldots, x_n)) \in \mathbf{F}_q[x_1, \ldots, x_n],$$

with $p_i(x_1, \ldots, x_n) \in \mathbf{F}_q[x_1, \ldots, x_n]$, for $i = 1, 2, \ldots, n$. The p_i are quadratic polynomials due to the choice of f and the fact that $x \mapsto x^q$ is a linear function of $\mathbf{F}_{q^n} \rightarrow \mathbf{F}_{q^n}$.

Note that f may be a permutation of $\mathbf{F}_{q^n}$, in which case each $a \in \mathbf{F}_{q^n}$ gives rise to precisely one solution $x \in \mathbf{F}_{q^n}$ to the equation $f(x) = a$. If f consists of more than one monomial in x, however, it seems to be difficult to choose f such that it is a permutation (cf. [5: Chapter 7] or [8]). Obviously, for any $a \in \mathbf{F}_{q^n}$ there are at most d solutions to $f(x) = a$, and often there are only a few. It is well known that solutions to $f(x) = a$ can be found in deterministic time polynomial in p, m, n, and d, and in expected time polynomial in $\log p$, m, n, and d, cf. [1: 17-26], [5: Chapter 4], [14], [15]. Some run times can be found in [7].

3 Hidden Field Equations for Encryption

In this paragraph we will describe a first version of HFE for encryption (i.e. the version with the easiest description).

We assume that the message M is represented as an n-tuple x over $\mathbf{F}_q$, where $\mathbf{F}_q$ as above is publicly known. (Thus, if $p = 2$, each message can be represented by nm bits.) Moreover, we assume that some redundancy has been included in the representation of each message, in such a way that the redundancy depends in a non-linear way on M. A nice way to do this is to make use of an error correcting code. If $p = 2$ we could also obtain x by concatenating the binary representation of M and the first 64 bits of $h(M)$, where h is a hash function such as MD5 or SHA, as long as the resulting x has at most nm bits.

Let s and t be two affine bijections $(\mathbf{F}_q)^n \rightarrow (\mathbf{F}_q)^n$, where $(\mathbf{F}_q)^n$ is regarded as an n-dimensional vector space over $\mathbf{F}_q$. Both s and t can be represented as n-tuples of polynomials in n variables over $\mathbf{F}_q$ of total degree 1. Using the function f from Section 2 and some representation of $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$ as in Section 2, the function $(\mathbf{F}_q)^n \rightarrow (\mathbf{F}_q)^n$ that assigns $t(f(s(x)))$ to $x \in (\mathbf{F}_q)^n$ can be written as

$$t(f(s(x_1, \ldots, x_n))) = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, x_2, \ldots, x_n)) \in \mathbf{F}_q[x_1, \ldots, x_n],$$

with $p_i(x_1, \ldots, x_n) \in \mathbf{F}_q[x_1, \ldots, x_n]$, for $i = 1, 2, \ldots, n$. The p_i are quadratic polynomials due to the choices of s, t, and f. Furthermore, given s, t, f and the way $\mathbf{F}_{q^n}$ is represented over $\mathbf{F}_q$, the polynomials p_i can efficiently be computed. The converse, however, seems to be hard, if s, t, and f are properly chosen. This leads to the following public key encryption scheme, which we call 'Hidden Field Equations' (HFE).

Secret key. A function f as in Section 2, two affine bijections s and t as above, and some way of representing $\mathbf{F}_{q^n}$ over $\mathbf{F}_q$. The latter may or may not be secret since changing the representation is equivalent to changing s or t; therefore, we may assume some fixed (and public) way of representing $\mathbf{F}_{q^n}$.

Public key. Polynomials p_i for $i = 1, 2, \ldots, n$ as above, computed using the secret key f, s, t. Furthermore, $\mathbf{F}_q$, the extension degree n and the way to add redundancy to a message are public.

Encryption. To encrypt the n-tuple $x = (x_1, \ldots, x_n) \in (\mathbf{F}_q)^n$ (representing the message M plus redundancy), compute the ciphertext

$$y = (p_1(x_1, \ldots, x_n), \ldots, p_n(x_1, x_2, \ldots, x_n)).$$

Decryption. To decrypt the ciphertext y, first find all solutions z to the equation $f(z) = t^{-1}(y)$ (cf. Section 2), next compute all $s^{-1}(z)$'s, and finally use the redundancy to find M from these.

Security considerations. We conclude this section with a few remarks concerning the security of HFE. We have restricted ourselves to the case where the characteristic p is equal to 2, even though HFE works for any small prime value p (unlike the Matsumoto-Imai scheme from [6], which only works for $p = 2$).

1. To avoid exhaustive search attacks we recommend that the message M consists of at least 64 bits and of at least 128 bits including redundancy.

2. In order to avoid the "affine multiple attack" that is described in the next section, we recommend to choose f of degree at least 17, but small enough to make decryption efficient. For computational examples of this attack we refer to the next section. Furthermore, to foil this attack it is necessary (but not sufficient) that f consists of at least two monomials in x: HFE with one monomial is equivalent to a Matsumoto-Imai algorithm, and can be attacked as described in [9]. We have done some Toy simulations of the vulnerability of HFE to the attack from [9] for $n = 13$. Though our tests enabled us to identify weak keys, they did not lead to a method to break HFE for well chosen keys. Details can be found in the extended version of this paper.

3. Some authentication algorithms (such as [12] or [13]) are proved to be as secure as a NP hard problem. (This is a very nice result of security but of course

this is not a proof of absolute security: a problem can be NP hard but easy in average, or easy with bad parameters, or difficult only with very large parameters). Can we also hope to prove that HFE is as secure as a NP hard problem? No: from a generalisation of a theorem given by G. Brassard in [2] we can prove that to recover a cleartext from an encrypted HFE Text is never an NP hard problem (if NP $\neq$ co NP). However this is not really a flaw of HFE but a property of almost all asymmetric encryption algorithms.

Idea of the proof. Let F be an asymmetric encryption algorithm with a secret key K and a public key k such that when the secret key K is given and when a value y is given it is always very easy to see if there is or not a cleartext x such that $y = F_k(x)$, i.e. such that y is the encryption of x by the algorithm F with the public key k. HFE, as all efficient encryption algorithms (such as RSA) has of course this property. Now let us consider the problem: "Is there an x such that $y = F_k(x)$?", where y is a given value. Then if the answer is "yes" x is a certificate that indeed the answer is "yes", i.e. it is easy to verify that the answer is "yes" if such an x is given (K is also another certificate). Moreover if the answer is "no" K is a certificate that indeed the answer is "no". So this problem is in NP $\cap$*co* NP. But (if NP $\neq$ *co* NP) there is no NP hard problem in NP $\cap$*co* NP. Similarly if from the secret key K we can compute easilly all the x such that $y = F_k(x)$, then the problem: "Is there an x such that $y = F_k(x)$ and $a \leq x \leq b$?", where a and b are two integers, is also in NP $\cap$*co* NP.So to recover a cleartext x from its corresponding cyphertext y can not be a NP hard problem. This shows that there is little hope to design any practical asymmetric encryption algorithm with a security proved to be based on a NP hard problem. It is also instructive to see that RSA may (or may not) be as secrure as the factorisation problem because the factorisation problem is in $NP \cap co\ NP$ (so is not a NP hard problem).

This result could suggest that when we have introduce a trapdoor in HFE, in order to have a cryptosystem usefull for encryption, we may have weaken the problem. This results shows also that the problem on with the security of HFE relies is not clearly shown (it can not be the general NP hard problem of solving randomly selected system of multivariate quadratic equations over $GF(2)$). However to recover a cleartext from its HFE ciphertext is still expected to be exponentally difficult when the HFE parameters are properly chosen.

4 The affine multiple attack

Introduction The "affine multiple attack" of the basic HFE that we will consider in this paragraph is a generalization of the main attack of [9] of the Matsumoto-Imai Algorithm. It is the only attack that we know against the basic HFE that can sometimes be much better than "quasi" exhaustive search on the cleartext (i.e. exhaustive search on most of the cleartext).

Principle of the attack Let f be a polynomial used in the basic HFE algorithm. By using a general algorithm (see for example [1] p. 25) we know that there are always some affine (in x) multiple $A(x,y)$ of the polynomial $f(x) - y$. (This means that $x \mapsto A(x,y)$ is an affine function and that each solution x of $f(x) = y$ is also a solution of $A(x,y) = 0$). For example in characteristic 2 the polynomial $A(x,y)$ will have only $1, x, x^2, x^4, x^8 \ldots, x^{2^j}, \ldots$ as monomials in x.

From now on we will assume for simplicity that the characteristic is 2.

Moreover, sometimes for such an affine multiple $A(x,y)$ all the exponents in y have small Hamming Weight in base 2. If this occurs, then the polynomial f will be a weak key for HFE.

More precisely if all the exponents in y have a Hamming Weight $\leq k$, then there will be an attack with a Gaussian reduction on $0(n^{1+k})$ terms (more precisely with about $\sum_{i=1}^{k} n^{1+i}/i!$ terms because we have about $n^{1+i}/i!$ terms of total degree i in the y_j variables, $1 \leq i \leq k$) where n is the number of bits of the message. This attack will work exactly as the attack of [9] for the Matsumoto-Imai Algorithm. The Gaussian reduction needed may be easier than a general Gaussian reduction but the complexity will be at worst in $0(n^{3k+3})$ and at least in $0(n^{1+k})$. Gaussian reductions with N terms are asymptotically in N^{ω} with $\omega < 2.376$ (cf. [3]). Moreover we can choose some x but not some y, so in the equations on which we need to do a Gaussian reduction we will have at least $O(n^{1+2k})$ unpredictible values. So it seems that more precisely the Gaussian reduction needed will be at most in $O(n^{(1+k)\omega})$ and at least in $O(n^{1+2k})$.

Example 1. Let $f(x) = x^{1+2^\theta}$. So $x^{1+2^\theta} = y$.

Then $x^{2^{2\theta}}.y = x.y^{2^\theta}$. So $A(x,y) = x^{2^{2\theta}}y + xy^{2^\theta}$ is an affine multiple of $f(x) + y$, and here all the exponents in y have a Hamming Weight ≤ 1. This leads to the attack with a Gaussian reduction on $0(n^2)$ terms described in [9].

Example 2. Let $f(x) = x^5 + x^3 + x = y$.

Then it is possible to prove that $y^3.x + (y^2 + 1)x^4 + x^{16} = 0$.

Here all the exponents in y have a Hamming Weight ≤ 2. So this leads to an attack of the HFE algorithm with a Gaussian reduction on $0(n^3)$ terms if this polynomial f is used. So this polynomial should not be used (it's a weak polynomial).

Example 3. Let $f(x) = x^9 + x^6 + x^5 + x^3 + x = y$.

Then the affine multiple $A(x,y)$ of $f(x)$ of degree 2^8 in x found by AXIOM is:

$$\begin{aligned}
&(y^{27}+y^{24}+y^{23}+y^{20}+y^{19}+y^{11}+y^8+y^7+y^4+y^3)x \\
&+(y^{27}+y^{25}+y^{21}+y^{20}+y^{15}+y^9+y^7+y^5+y^4+y^3)x^2 \\
&+(y^{28}+y^{26}+y^{25}+y^{20}+y^{18}+y^{16}+y^{14}+y^9+y^8+y^6+y^4+1)x^4 \\
&+(y^{22}+y^{21}+y^{18}+y^{16}+y^{15}+y^{14}+y^{13}+y^{10}+y^8+y^7)x^8 \\
&+(y^{25}+y^{22}+y^{21}+y^{19}+y^{17}+y^{12}+y^{11}+y^6+y^5)x^{16} \\
&+(y^{23}+y^{20}+y^{19}+y^{18}+y^{17}+y^{16}+y^{15}+y^{14}+y^{13}+y^{11}+y^{10}+y^9+y^8+y^6+y^5)x^{32} \\
&+(y^{18}+y^{17}+y^{14}+y^{11}+y^{10}+y^9+y^6+y^3)x^{64} \\
&+(y^{13}+y^{11}+y^5+y^4+y^3)x^{128} \\
&+x^{256}.
\end{aligned}$$

In $A(x,y)$ the largest Hamming Weight of the exponents in y is 4.
So this leads to an attack with a Gaussian reduction on $0(n^5)$ terms if this polynomial is used. This attack will need a lot of power but may be feasible. (For example if $n = 64$ it will need Gaussian reduction on 2^{25} variables ($\simeq n^5/4!$) and if $n = 128$ it will need Gaussian reduction on 2^{30} variables...). So we do not recommend to use this function f.

Example 4. Let $f(x) = x^{12} + x^8 + x^4 + x^3 + x^2 + x = y$.
Then one of the affine multiple of $f(x)$ found by axiom is:

$$\begin{aligned}
&x^{256} + (y^{16}+y)x^{64} + (y^8+y^5+y^2)x^{16} + (y^3+1)x^4 \\
&+yx + y^{16} + y^8 + y^5 + y^3 + y^2 = 0.
\end{aligned}$$

Here all the exponents in y have a Hamming Weight ≤ 2.
So this polynomial f should not be used for HFE.
Since the degree of f was not so small (it was 12), and since f had a lot of monomials (6), this example shows that the affine multiple attack has to be taken seriously: it is not always obvious whether it works or not.

Example 5. Let $f(x) = x^{17} + x^9 + x^4 + x^3 + x^2 + x = y$.
With AXIOM, we have computed the least affine multiple $A(x,y)$ of $f(x) + y$ (it took us two days on a workstation).
In $A(x,y)$ all the exponents in y are ≤ 3840, and the exponent with the largest Hamming Weight as a Hamming Weight of 11.
So this affine multiple leads to an attack of HFE with this polynomial f with a Gaussian reduction on $0(n^{12})$ terms, where $n \geq 64$. (For $n = 128$ it will need Gaussian reduction on 2^{58} terms because $2^{58} \simeq n^{12}/11!$).
Since this attack is completely impracticable, this polynomial f resists to the "affine multiple attack" and may be a strong polynomial for HFE.

Example 6. Let $f(x) = x^{17} + x^{16} + x^5 + x = y$.
With AXIOM (also after two days of computations) we have computed the least affine multiple $A(x,y)$ of $f(x) + y$. In $A(x,y)$ the exponents with the largest Hamming Weight have a Hamming Weight also of 11.
So this function may also be a strong polynomial for HFE.

Note: What is nice with this function is that this function is not only quadratic over $\mathbf{F}_2$ but also quadratic over $\mathbf{F}_4$. (So the public computations will be easier with this function).

Asymptotic complexity For large d, and for most of the polynomials f of degree d, the complexity of the affine multiple attack of the basic HFE with this polynomial f is expected to be in $0(n^{0(d)})$.
So, if $d = 0(n)$ the complexity of the attack is expected to be exponential in n. Moreover, $d = 0(\ln n)$ is expected to be sufficient to avoid all polynomial attacks.

Conclusion The affine multiple attack is very efficient for some very special polynomials. However when the degree of f is ≥ 17 and when f is well chosen, this attack is expected to fail completely.

Note For easier computations, we have chosen in all the examples the constant terms in the monomials of f equal to 0 or 1.
Of course this is not an obligation and any elements of the extension field can be chosen.

5 HFE variations

HFE plus and minus some p_i equations. The polynomials $(p_1, \ldots, p_n)$ of the HFE Algorithm of paragraph 3 gives y from x. Since there is some redundancy in x it may be possible to recover x from y without some of these polynomials. For example when p_{n-1} and p_n are omitted it may still be possible to recover x from y: we will just compute the 2^{2m} possibilities and find the good x thanks to the redundancy in x. When m is very small, for example when $m = 1$ or 2 this is clearly feasible.
So we can imagine that just $p_1, \ldots p_{n-2}$ are public.

Note. This idea of ommiting some polynomial p_i can also be done in the original Matsumoto-Imai scheme (instead of HFE). However this is not recommended: in the extended version of this paper we give some ideas for the cryptanalysis of such a scheme.

HFE with multivariate field equations for encryption. Here the idea is to change the description of the function f given in paragraph 2. We can notice that what we need for f is that:
1. In a basis, f is a multivariate quadratic function.
2. For any value a, it is easy to find all the x so that $f(x) = a$.
3. f is a function with inputs and outputs with at least 64 bits (the reason for this is that we can not have small "branches" in the algorithm: we will give more details about this in the next paragraph).
The solution given in paragraph 2 was to choose for f a polynomial in only one variable x over $\mathbf{F}_{q^n}$ so that, in a basis, f is a multivariate quadratic function.

In the extended version of this paper we present some different candidates for f: polynomials in two variables x_1 and x_2, or more. An efficient algorithm of resolution of $f(x) = a$ (for example with Gröbner basis or with something else) will be hidden by the two affine functions s and t.

HFE with more than one branch. In the original Matsumoto-Imai Algorithm [6] the values are split in different branchs after the first affine transformation s. We could also imagine to do this in a HFE scheme. However, in the extended version of this paper we show that if the branchs are small then it is always easy to attack the scheme by detecting and isolating the small branchs. So we do not recommend to use more than one branch.

HRE: Hidden Rings Equations. In paragraph 2 we said that the field $\mathbf{F}_{q^n}$ is typically $\mathbf{F}_q[x]/(g(x))$, where $g(x) \in \mathbf{F}_q[x]$ is irreductible of degree n.
If $g(x)$ is not irreductible, then $\mathbf{F}_q[x]/(g(x))$ will then not be a finite field, but a finite ring. In such a space the resolution of $f(x) = y$, where f is a univariate polynomial is still feasible. For example the linearized polynomial algorithm still works. So we can design an asymetric scheme in such a space exactly as HFE in the finite field $\mathbf{F}_{q^n}$.

HFE with public polynomials of degree ≥ 3. Of course we can also choose for f a polynomial with some exponents in x of Hamming Weight still small but ≥ 3. A very important subcase, from a practical point of view, if when this function is $f(x) = x^{1+q^\theta+q^\varphi}$, i.e. with only one monomial and Hamming Weight 3. The study of these functions is one of the main subject of [10].

Concatenation of two basic HFE or HRE for fast decryptions. Let x be the cleartext. Let $y_1 = HFE_1(x)$ be the encryption of x with a first HFE encryption with secret affine functions s_1 and t_1. Let $y_2 = HFE_2(x)$ be the encryption of x with another HFE, such that HFE_1 and HFE_2 have different polynomials f_1 and f_2 and independent secret affine functions t_1 and t_2, but the same extension field $\mathbf{F}_{q^n}$, and the same secret affines functions $s_1 = s_2$. Then let $y_1||y_2$ be the encryption of x, where $||$ is the concatenation function.
The main advantage of this scheme is that decryption with the secret keys may be very fast, as we will see now. From y_1 and y_2, $f_1(a)$ and $f_2(a)$ will be obtained, and then *GCD*$(f_1(a), f_2(a))$ will be computed. Then from this *GCD* the value of a will be obtained with one of the classical algorithm of resolution of equation. Then $x = s_1^{-1}(a)$ will be obtained.
In average the time of computation of $GCD(f_1(a), f_2(a))$ is expected to be dominant. This time is $\leq O(d^2 n^2)$, where $d = \sup(d_1, d_2)$. So if d_1 and d_2 are not too large decryption will really be very fast (and much faster than in the basic HFE).
For example the complexity of decryption may be $\leq O(n \ln^3 n)$ for well chosen f_1 and f_2 with degree $(f_1) \leq O(\ln n)$ and degree $(f_2) \leq O(\ln n)$.
However it is not recommended generally in cryptography to encrypt the same message twice by two different encryptions. Moreover this is generally particulary

not recommended when the two encryptions are not independents. So if this variation is really used we recommend to be extra-careful in the choice of the polynomials f_1 and f_2. For example not only f_1 and f_2 should avoid the "Affine multiple attack", but also $f_1 + f_2$.

6 HFE in signature or Authentication

All encryption algorithm can also be use as an authentication algorithm: the verifier will encrypt a challenge and ask for the cleartext. So HFE can be use for authentications. Moreover HFE can also be slightly modified in order to give asymmetric signatures. We will now give two examples of such transformations. In the first example the signatures will have 160 bits, and in the second example the signature will have about 128 bits. However in these examples the time needed to compute a signature is not constant: some messages may be much easier to sign than some others.

Example 1

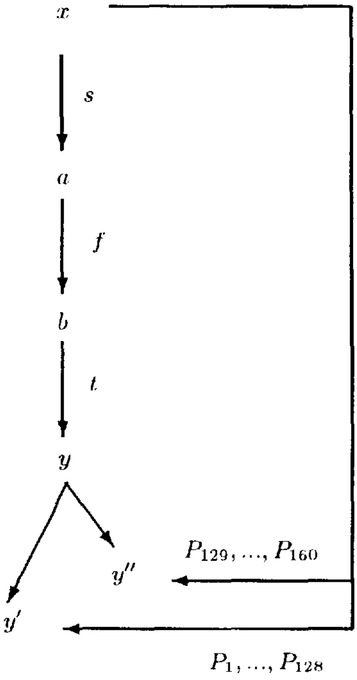

Fig. 1. Example 1 of HFE in signature. x : the signature (160 bits). y' : the hash to sign (128 bits). $P_1, \ldots, P_{128}$ are public. $P_{129}, \ldots, P_{160}$ are secrets.

Let us consider an HFE algorithm, as described in the next paragraphs, with x and y of about 128+32=160 bits.
Let $p_1, \ldots, p_n$ be the n public polynomials that give y from x, with $n = 160$ and $\mathbf{F}_q = \mathbf{F}_2$ for example.
If only p_1 to p_{128} of these polynomials are public (the over are secret), then the polynomials $p_1, \ldots p_{128}$, give a value z of 128 bits from a value x of 160 bits.
In our algorithm here z is the hash of a message to sign and x will be the signature of z. When z is given, then with the secret polynomials p_{129} to p_{160} and the other secret values we will be able to find a value x so that from this x, the polynomials $p_1, \ldots, p_{128}$ will give exactly the value z.
For this we will padd z with 32 extra bits and try to find an x with a decryption of the HFE algorithm. If it fails we try with another padding until we succeed.
In figure 1 we illustrate such a use of HFE in signature.

Example 2

Computation of the signature In this example 2 to sign a message M there will be three steps.

Step 1. We generate a small integer R with no block of numbers with 10000 in its expression in base 2 (for example $R = 0$ to start).
Step 2. We compute $h(R||10000||M)$ where h is a public collision free hash function with an output of 128 bits (for example h is the MD5 algorithm).
Step 3. We consider an HFE algorithm (as in paragraph 3) with values x and y of 128 bits.
If we take $y = h(R||10000||M)$, then we can (with the secret key) try to find a cleartext x so that $HFE(x) = y$.
If we succeed, then $R||x$ will be the signature of M.
If we do not succeed (because since HFE is not a permutation some value y have no corresponding x) then we try again at Step 1 with another R (for example with the new R equal to the old $R + 1$ if this new R has no block of 10000 in base 2).

Verification of a signature The message M and a signature $R||x$ of M is given. First, we separate R and x (since x has a fix length of 128 bits this is easy). Then we compute $h(R||10000||M)$ and $HFE(x)$ and the signature is valid if $h(R||10000||M) = HFE(x)$.

Length of the signature In this example 2 the length of the signature is not fixed. However in average R will be very small so that the signature $R||x$ will have in average just a few more than 128 bits.

Note. Of course the pattern 10000 is just an example and another pattern P can be chosen. More precisely the property that we want is that from $R||P||M$ we can recover R and M when we know that R do not have the pattern P. (So the pattern will have at least one 1 and one 0).

7 The Isomorphism of Polynomials (IP) authentication and signature scheme

7.1 Introduction

We will now present a new authentication scheme called "Isomorphism of Polynomials" (IP).
IP authentications have a few nice properties:
- It is proved zero-knowledge.
- We know exactly the problem on which the security of the scheme relies.
- The scheme is very symmetric, and the design of the scheme is very similar to the well known "Graph Isomorphism Authentication scheme" (cf. [11] p. 88-89 for example).
- No hash functions are needed.
- IP illustrates the fact that if in HFE the function f is public the HFE scheme may be still secure.

However if all these nice properties show that IP is a scheme of theoretical interest IP may not be as practical in Authentication as the schemes of [12] or [13] or as HFE. (Essentially because of the large number of bits to exchange or because of the lenght of the public key). Moreover HFE can be used for authentication, signatures, or encryption, and IP is just for authentication or signatures.

7.2 The Isomorphism of Polynomials Problem with two secrets s and t

Let u and n be two integers. Let $\mathbf{F}_q$ be a finite field.
Let A be a public set of u quadratic equations with n variables $x_1, \ldots, x_n$ over the field $\mathbf{F}_q$. We can write all these equations like this:

$$y_k = \sum_i \sum_j \gamma_{ijk} x_i x_j + \sum_i \mu_{ik} x_i + \delta_k, \quad \text{for } k = 1, \ldots u \tag{1}$$

Now let s be a bijective and affine transformation of the variables $x_i, 1 \leq i \leq n$, and let t be a bijective and affine transformation of the variables $y_k, 1 \leq k \leq u$. Let $s(x_1, \ldots, x_n) = (x'_1, \ldots, x'_n)$, and $t(y_1, \ldots, y_u) = (y'_1, \ldots, y'_u)$.
From (1) we will obtain k equations that gives the y'_k values from the x'_i values like this:

$$y'_k = \sum_i \sum_j \gamma'_{ijk} x'_i x'_j + \sum_i \mu'_{ik} x'_i + \delta'_k, \quad \text{for } k = 1, \ldots u. \tag{2}$$

Let B be the set of these u equations. We will say that A and B are "isomorphic", i.e. there is a double bijective and affine transformation that gives B from A. And we will say that (s, t) is an "isomorphism" from A to B. The "Isomorphism of Polynomials Problem" is this problem: when A and B are two public sets of u quadratic equations, and if A and B are isomorphic, find an isomorphism (s, t) from A to B.

Example. If $u = n$ no polynomial algorithms to solve this problem are known. If such an algorithm were found then it would give us a way to find the keys of the Matsumoto-Imai algorithm (and not only a way to decrypt most of the messages). So it would give us a new, and more powerful, attack on the Matsumoto-Imai Algorithm. Moreover if such an algorithm were found then in HFE it would be essential for security to keep f secret. On the contrary as long as no such algorithm is found HFE may be still secure if f is public.

Note. We could think to proceed like this in order to find s and t: to introduce the matrix of s and t values and to formaly identify the equations (1) and (2). However we will obtain like this some equations of total degree three in the values of s and t and the general problem of solving equations of degree $\geq$ two in a finite field is NP hard. So this idea does not work.

7.3 The IP authentication scheme with two secrets s and t

Public: Two isomorphic sets A and B of u quadratic equations with n variables over a field $\mathbf{F}_q$.
Secret: An isomorphism (s, t) from A to B.

Notations The equations of A are the equations (1) of paragraph 7.2, they give the y_k values from x_i values, and the equations of B are the equations (2) of paragraph 7.2, they give the y'_k values from the x'_i values.
Let us assume that Alice knows the secret (s, t) and that Alice wants to convince Bob of this knowledge, without revealing her secret. Alice and Bob will follow this protocol:

Step 1. Alice randomly computes a set C of equations isomorphic to A.
For this, she randomly computes an affine bijection s' of the values $x_i, 1 \leq i \leq n$, and an affine bijection t' of the variables y_k, $1 \leq k \leq u$.
The u equations of C are u equations like this:

$$y''_k = \sum_i \sum_j \gamma''_{ijk} x''_i x''_j + \sum_i \mu''_{ik} x''_i + \delta''_k, \quad \text{for } k = 1, \dots u. \qquad (3)$$

- s gives the transformation $x \longrightarrow x'$.
- t gives the transformation $y \longrightarrow y'$.
- s' gives the transformation $x \longrightarrow x''$.
- t' gives the transformation $y \longrightarrow y''$.

Step 2. Alice gives the set C of equations (3) to Bob.
Step 3. Bob asks Alice either to

(a) Prove that A and C are isomorphic.
(b) Prove that B and C are isomorphic.

Moreover Bob choose to ask (a) or (b) randomly with the same probability 1/2.

Step 4. Alice complies.

If Bob ask (a), then she reveals s' and t'.

If Bob ask (b), then she reveals $s' \circ s^{-1}$ and $t' \circ t^{-1}$ (i.e. the transformations $x' \rightarrow x''$ and $y' \rightarrow y''$).

It is easy to prove that this protocol is zero-knowledge and that if somebody doesn't know an isomorphism (s, t) from A to B the probability to successfully pass the protocol is at most 1/2.
So if Alice and Bob repeat steps (1) to (4) N times, the probability of success will be at most $1/2^N$.

7.4 Parameters

Analogous to the problem of finding the secret affine transformations s and t of HFE when f is public or of the Matsumoto-Imai Algorithm, we could have $u = n = 64$ or 128 and $\mathbf{F}_q = \mathbf{F}_2$ for example in a IP authentication scheme. However more practical values may be sufficient for security.
In the extended version of this paper we give some comments about more practical values.

7.5 The IP problem with one secret s

Let n be an integer. Let $\mathbf{F}_q$ be a finite field. Let A be one public cubic equation with n variables $x_1, \ldots, x_n$ over the field $\mathbf{F}_q$. (Here in A we have only one equation, but of degree 3 and not 2). We can write this equation (A) like this:

$$\sum\sum\sum \gamma_{ijk} x_i x_j x_k + \sum\sum \mu_{ij} x_i x_j + \sum \alpha_i x_i + \delta_0 = 0. \qquad (A).$$

Now let s be a bijective and affine transformation of the variables $x_i, 1 \leq i \leq n$. Let $s(x_1, \ldots, x_n) = (x'_1, \ldots, x'_n)$.
From (A) we will obtain one equation (B) in x'_i like this:

$$\sum\sum\sum \gamma'_{ijk} x'_i x'_j x'_k + \sum\sum \mu'_{ij} x'_i x'_j + \sum \alpha'_i x'_i + \delta_0 = 0 \qquad (B).$$

We will say that (A) and (B) are "isomorphic", i.e. there is a bijective and affine transformation that gives B from A. And we will say that s is an "isomorphism" from A to B. The "Isomorphism of Polynomials Problem" is now this problem: when A and B are two public sets of such equations, and if A and B are isomorphic, find an isomorphism s from A to B.

Note. For equations of total degree ≥ 3, we know no polynomial algorithm to solve this IP problems. However for equations of total degree 2 there is a polynomial algorithm to solve the problem, because there is a "canonical" representation of each quadratic equations over a finite field (cf. [5], chapter 6 for example). If (A) and (B) are isomorphic, then the two canonical representations of (A) and (B) will be easilly found, will be the same, and this will give the isomorphism from (A) to (B). This is the reason why we have chosen equations of degree ≥ 3.

7.6 The IP authentication scheme with one secret s

The IP problem with one secret s can easilly be use to design an authentication scheme in the same way we used the IP problem with two secrets s and t (more details are given in the extended version of this paper).

7.7 Less computations with larger public keys

Instead of only two public isomorphic equations (A) and (B), less us now assume that we have k public isomorphic equations $(P_1), (P_2), \dots (P_k)$.
We denote by $x_i^{(1)}$ the variables of $(P_1), \dots$ and by $x_i^{(k)}$ the variables of (P_k). And we denote by s_j the secret affine transformation from $x^{(1)}$ to $x^{(j)}$, $2 \leq j \leq k$. (So all the k equations are isomorphic to (P_1), so each couple of these equations are isomorphic).
Of course we can assume if we want that all the secret affine transformations $s_j, 2 \leq j \leq k$, are computed from one small secret K, for example K is a secret DES key and the matrix of the s_j are obtained by some computations of DES_K. So the public key is larger, since we have k equations (P_j), but the secret key can be still small.
The authentication now proceed like this:

Step 1. Alice randomly computes, as usual, one equation C isomorphic to P_1.
Step 2. Alice gives this equation C to Bob.
Step 3. Bob randomly chose a value u, $1 \leq u \leq k$, and ask Alice to prove that C and P_u are isomorphic.
Step 4. Alice complies.

It is still easy to prove that this protocol is zero-knowledge and that if somebody doesn't know any isomorphism s from one (P_i) to one (P_j), $i \neq j$, then the probability to successfully pass the protocol is at most $1/k$.
So if Alice and Bob repeat steps (1) to (4) N times, the probability of success will be at most $1/(k^N)$.

7.8 IP for asymmetric signatures

The Fiat-Shamir authentication scheme and the Guillou-Quisquater authentication scheme can be transformed in signature scheme by using a now classical transformation by introducing hash function. This transformation works also very well here for the IP algorithm.
Let M be the message to sign. The signature algorithm is this one:

Step 1. Alice randomly computes λ equations C_i isomorphic to P_1.
Step 2. Alice computes hash $(M||C_1|| \dots C_\lambda)$, where $||$ is the concatenation function, and hash a public hash function sufficiently large such that the first bits of output can give λ values $e_1, \dots, e_\lambda$, where each e_i is a value between 1 and k.

Step 3. Alice computes the λ isomorphisms $t_i, 1 \leq i \leq \lambda$, such that each t_i is an isomorphism from C_i to P_{e_i}.

The signature on M by Alice is then (T, E) where T is the vector $(t_1, t_2, \ldots t_\lambda)$ and E is the vector $(e_1, e_2, \ldots e_\lambda)$.

To verify this signature, Bob proceeds like this:
Step 1. Bob computes $C_1, \ldots, C_\lambda$ such that $t_i, 1 \leq i \leq \lambda$ is an isomorphism from C_i to P_{e_i}.
Step 2. Bob checks that the first bits of hash $(M||C_1|| \ldots C_\lambda)$ are the entries e_i of E.

7.9 Numerical examples of IP signatures with one secret s

In signature we must have $k^\lambda \geq 2^{64}$ for security.
It is not clear what value of n should be taken, but we suggest $n \geq 16$ if $K = \mathbf{F}_2$. With $K = \mathbf{F}_2, n = 16, \lambda = 16$ and $k = 16$ then the lenght of the public key is 1120 bytes and the lenght of the signature is about 4128 bits. With $K = \mathbf{F}_2, n = 16, \lambda = 4$ and $k = 2^{16}$ then the lenght of the public key is $k.16.15.14/3!$ bits = 4,4 Mo. This is huge but can be store in a hard disc of a Personal computer, and the lenght of the signature is $\simeq 4.(16 + 16.16) = 1088$ bits.

8 Conclusion

We have designed two new classes of Algorithms: HFE and IP. These algorithms are based on multivariate polynomials over a finite field of total degree two. One interesting point of HFE is that these algorithms can lead to very short asymmetric signatures (128 bits for example). Similarly they can encrypt messages by blocks whith very short blocks (128 bits blocks for example).
Another interesting point of these algorithms is that their security do not depend on factorisation or discret log, and very few algorithms for encryption or signatures in asymmetric cryptography are known that do no rely on these problems.
However a lot of problems are still open, for example:
Are these algorithms really secure ?
Is it possible to design strong HFE, with public polynomials of degree two and a secret function f with two or more monomials, that are also permutations ?
Is it possible to solve a general system of multivariate quadratic equations over $GF(2)$ much more quickly than with a quasi exhaustive search ?

Acknowledgments

I want to thank Isabelle Guerrin-Lassous for doing the simulations of paragraph 3.2. I want also to thank Daniel Augot for doing the simulations of paragraph 4. However Daniel Augot told me that he does not want to endorse the security of HFE... Finally I want to thank Arjen Lenstra for all the time he has spend to improve the redaction of this paper.

References

1. F. BLAKE, X. GAO, R. MULLIN, S. VANSTONE and T. YAGHOOBIAN, *"Application of Finite Fields"*, Kluwer Academic Publishers.
2. G. BRASSARD, *"A note on the complexity of cryptography"*, IEEE Tran. Inform. Theory, Vol. IT-25, pp. 232-233, 1979.
3. D. COPPERSMITH and S. WINOGRAD, *"Matrix Multiplication via Arithmetic Progressions"*, J. Symbolic Computation, 1990, Vol. 9, pp. 251-280.
4. M. GAREY, D. JOHNSON, *"Computers and intractability, A Guide to the Theory of NP-Completeness"*, FREEMAN.
5. R. LIDL, H. NIEDERREITER, *"Finite Fields"*, Encyclopedia of Mathematics and its applications, Volume 20, Cambridge University Press.
6. T. MATSUMOTO and H. IMAI, *"Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption"*, EUROCRYPT'88, Springer Verlag 1988, pp. 419-453.
7. A. MENEZES, P. VAN OORSCHOT ans S. VANSTONE, *"Some computational aspects of root finding in $GF(q^m)$"*, in Symbolic and Algebraic Computation, Lecture Notes in Computer Science, 358 (1989), pp. 259-270.
8. Gary L. MULLEN, *"Permutation Polynomials over Finite Fields"*, in "Finite Fields, Coding Theory, and Advances in Communications and Computing", Dekker, Volume 141, 1993, pp. 131-152.
9. J. PATARIN, *"Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88"*, CRYPTO'95, pp. 248-261.
10. J. PATARIN, *"Asymmetric Cryptography with a Hidden Monomial"*, available but not yet published paper.
11. B. SCHNEIER, *"Applied Cryptography"*, John Wiley and Sons, first edition.
12. A. SHAMIR, *"An efficient Identification Scheme Based on Permuted Kernels"*, CRYPTO'89, pp. 606-609.
13. J. STERN, *"A new identification scheme based on syndrome decoding"*, CRYPTO'93, pp. 13-21.
14. P. VAN OORSCHOT and S. VANSTONE, *"A geometric approach to root finding in $GF(q^m)$"*, IEEE Trans. Info. Th., 35 (1989), pp. 444-453.
15. J. VON ZUR GATHEN and V. SHOUP, *"Computing Frobenius maps and factoring polynomials"*, Proc. 24th Annual ACM Symp. Theory of Comput., ACM Press, 1992.