A Family of Weak Keys in HFE (and the Corresponding Practical Key-Recovery)

Charles Bouillaguet[1],
Pierre-Alain Fouque[1], Antoine Joux[2,3], and Joana Treger[2,4]

[1] Ecole Normale Supérieure
{charles.bouillaguet, pierre-alain.fouque}@ens.fr
[2] Université de Versailles-Saint Quentin
[3] DGA
antoine.joux@m4x.org
[4] ANSSI
joana.treger@ssi.gouv.fr

Abstract. The HFE (Hidden Field Equations) cryptosystem is one of the most interesting public-key multivariate scheme. It has been proposed more than 10 years ago by Patarin and seems to withstand the attacks that break many other multivariate schemes, since only subexponential ones have been proposed. The public key is a system of quadratic equations in many variables. These equations are generated from the composition of the secret elements: two linear mappings and a polynomial of small degree over an extension field. In this paper we show that there exist weak keys in HFE when the coefficients of the internal polynomial are defined in the ground field. In this case, we reduce the secret key recovery problem to an instance of the Isomorphism of Polynomials (IP) problem between the equations of the public key and themselves. Even though for schemes such as SFLASH or C^* the hardness of key-recovery relies on the hardness of the IP problem, this is normally not the case for HFE, since the internal polynomial is kept secret. However, when a weak key is used, we show how to recover all the components of the secret key in practical time, given a solution to an instance of the IP problem. This breaks in particular a variant of HFE proposed by Patarin to reduce the size of the public key and called the "subfield variant". Recovering the secret key takes a few minutes.

Key words: Cryptanalysis, multivariate cryptography, HFE, weak keys, Gröbner Bases.

1 Introduction

Multivariate cryptography is interesting from several points of view. First of all, it is based on a hard problem, namely solving system of multivariate equations, for which there only exist generic algorithms whose complexity is exponential in the worst case. Then, it has been proposed as an alternative to the RSA cryptosystem since there is no quantum algorithms for this problem. Finally, it is also appealing since the public operation does not require computations with large integers, and no crypto-processor is needed.

The HFE cryptosystem has been proposed in 1996 by Patarin in [28] in order to avoid his attack on the Matsumoto-Imai cryptosystem [23, 27]. This last scheme has also been called C^* and basically hides the power function $X \mapsto X^{1+q^\theta}$ in an extension field of degree n over $\mathbb{F}_q$, using two secret linear bijections S and T. In order to invert it, it suffices to remark that this power function, as the RSA power function, can be easily inverted provided $1 + q^\theta$ is invertible modulo $q^n - 1$. In [28], Patarin proposed to change the internal *known* monomial into a *secret* polynomial $\mathbf{f}$ of small degree. The legitimate user can still easily invert the public key since she knows S and T, and can invert the small degree polynomial using the Berlekamp algorithm for instance.

1.1 Related Works

From the adversary point of view, the action of S and T transforms the secret internal polynomial into a very sparse univariate polynomial of very high degree, as shown for instance by Kipnis and Shamir in [22].

A possible decryption attack would consist in inverting or factorizing this polynomial. However, there are no efficient algorithms to perform these tasks (an attempt can be found in [35]), and merely deciding the existence of roots is in fact NP-complete (*cf.* [22]).

HFE belongs to the category of public-key cryptosystems based on the hardness of computing a *functional decomposition*: given the composition of two functions f and g, can one identify the two components? Other examples include C^*, SFLASH [30], FAPKC [36], 2R [32] and McEliece [24]. With the exception of the latter, the former have all been broken because computing a functional decomposition was not as hard as expected. In the context of HFE, computing such a decomposition is related to decomposing the univariate representation of the public key, in order to recover the secret internal polynomial $\mathbf{f}$ as well as polynomial representations of S and T. Computing polynomial decompositions is a simple and natural mathematical problem which has a long history, going back to the works of Ritt and Ore in 1922 and 1930 respectively [34, 26]. Today, polynomial decomposition algorithms exist for some classes of polynomials over finite fields [37, 38], but no such algorithm is applicable to HFE. One step of the attack presented in this article amounts to computing a polynomial decomposition, and makes use of Gröbner bases.

The complexity of existing attacks, which all amount to solving systems of quadratic equations, depends on the degree d of the secret internal polynomial. When this degree is fixed, their complexity is polynomial in the security parameter n, although the exponent can be ridiculously large. In order for decryption to be polynomial, d must grow at most polynomially in n, and in that case the attacks are no longer polynomial. We consider this setting to be the most natural one to compare the asymptotic complexity of these attacks.

A simple decryption attack against HFE consists, given a ciphertext, in trying to solve the equations given by the public key. In 2003, Faugère and Joux experimentally showed that the HFE equations are not random systems of multivariate equations, because computing a Gröbner basis for these equations is much easier than the corresponding problem with random quadratic equations [16]. This allowed a custom implementation of the F5 algorithm [15] to break the first HFE challenge, for which the public key has 80 quadratic equations in 80 unknowns over $\mathbb{F}_2$. Later, Granboulan *et al.* [20] showed that specific algebraic properties of the HFE equations make the complexity of inverting HFE subexponential, in $\mathcal{O}\left(\exp\left(\log^2 n\right)\right)$.

In general, the hardness of recovering the secret key of HFE from the public key is unrelated to the Isomorphism of Polynomials (IP) problem [28], unless the internal polynomial is made public. A key recovery attack in the usual case where this polynomial is secret was presented in [22] and turns the problem of recovering T into an instance of the MinRank problem, the decisional version of which is NP-Complete [6]. Solving this instance of MinRank can be done by solving an overdetermined system of about n^2 quadratic equations in about $n \cdot \log d$ variables. The complexity of solving these equations is subexponential in $\mathcal{O}\left(\exp\left(\log^3 n\right)\right)$. This is too high to be practical, even for parameters corresponding to the HFE challenge that was broken.

These results show that HFE is not as robust as expected. However, can we consider HFE really broken? Is it still a viable alternative to RSA?

The cryptographic community often perceives HFE as broken, because of the practical attacks on some instances, and vastly lost both trust and interest in it. We would like to argue that the situation of HFE is slightly more complex. The complexity of some Gröbner basis algorithms, like F5 [15] is better understood [1] and allows to estimate the complexity of the decryption attacks, which remains relatively high for general instances. Moreover, standard modifications – such as removing some equations from the public key– destroy the algebraic structure presented by public key and that was exploited by Gröbner basis algorithms. HFE with Removed public equations is often called HFE$^-$, and suitable for a signature scheme. No attack faster than exhaustive search are known against HFE$^-$. In particular, the second HFE cryptanalytic challenge, with removed public equations, is currently far from being broken. Furthermore, it is suggested in [11] (based on experimentations) that the subexponential behavior of the Gröbner basis computation is mostly due to the fact that the computations are performed over $\mathbb{F}_2$, and that over odd-characteristic fields, computing a Gröbner basis of the public-key is no longer subexponential, but plainly exponential. This would mean that even when HFE is used for encryption, there are non-broken parameters.

All in all, HFE is comparatively in better shape than the SFLASH signature scheme for which polynomial time algorithms are known both to invert [13, 12] and to recover equivalent secret keys [18]. These attacks against SFLASH exploit the fact that multiplication matrices commute in some way with the internal monomial[1]. Then, it is possible to recover conjugates of the multiplications by the secret matrix S using simple linear algebra on the differential of the public key [18]. However, for general HFE, the multiplications no longer commute with the secret polynomial. Another issue is that we also need to recover the internal secret polynomial.

1.2 Our Results

In this paper, we consider the key recovery problem on a class of *weak keys* for HFE. As opposed to the decryption attack of Faugère and Joux [16], we recover an equivalent representation of the secret key that subsequently allows to inverse the trapdoor with the same complexity as the legitimate user. The weak instances we attack are defined by using an internal polynomial with coefficients in the ground field and not in the extension field as it was originally specified, or instances that are reducible to these specific ones (by considering equivalent transformations S and T, see section 3.1). Some instances belonging to this category were proposed by Patarin himself in [29] (an extended version of [28]) with the aim of reducing the size of the HFE public key (the so-called "subfield" variant). However, notice that the family of weak keys described here does not reduce to this subfield variant, and choosing the coefficients of the secret polynomial in the base field can seem rather natural. While in general, the hardness of the key-recovery does not depend on the hardness of the IP problem, we show that key recovery can be reduced to an instance of the IP problem, and that the solutions of this problem allow us to efficiently recover all the secret elements (or equivalent data). The latest IP algorithms allows to solve the instances in practice for realistic parameters set. To mount our attack, as in the SFLASH case [12], we try to find a commutation property to gain information about the secret key. In our attack, since multiplications no longer commute, we instead use the Frobenius map.

Coming back to the subfield variant, other schemes, including UOV [21] for instance, also have subfield variants, and the default in the design of an older version of SFLASH (v1) was to choose the secrets in a subfield. These schemes, or their subfield variants have all been broken: SFLASH v1 was attacked by Gilbert and Minier in [19], and subfield-UOV was shown to be insecure as well [4]. Although SFLASH and HFE share a similar structure, the Gilbert-Minier attack against SFLASH v1 cannot be applied to subfield-HFE, since it is based on Patarin's attack against C^*. Because this latter attack has no equivalent for HFE, there is no known attack against the subfield variant of HFE.

As mentioned above, the complexity of nearly all existing attacks depends on the degree of the internal secret polynomial. Even the most concrete and realistic threat, namely computing a Gröbner basis of the public-key, will become irrealistic if this degree is chosen high enough (a

[1] SFLASH is based on C^* and has a single internal monomial.

drawback is that decryption then becomes slower). A nice feature of the attack presented in this paper is that its asymptotic complexity is only marginally affected by the degree of the internal polynomial. As such, it be applied *in practice* to HFE instances on which existing attacks would be completely intractable. We also argue that under standard conjectures on the complexity of Gröbner basis computation, it is possible to establish that the complexity of our remains polynomial when the degree of the internal polynomial grows polynomially with n.

1.3 Organization of the Paper

Section 2 gathers some mathematical results, as well as basics on the HFE cryptosystem. In subsection 2.3, we give known results on the problem of finding isomorphisms of polynomials, that we need to mount our attack. Then, we describe our attack on the specific instances of HFE mentioned before in section 4. Finally, in section 5, to illustrate the attack, we show that we can break in practice a wide range of realistic parameters, including the ones proposed by Patarin for the "subfield" variant.

2 About HFE

2.1 Mathematical Background

Extension Fields and Vector Spaces. Let $\mathbb{K}$ be the finite field with q elements and $\mathbb{L}$ an extension of $\mathbb{K}$ of degree $n > 1$. $\mathbb{L}$ is isomorphic to $\mathbb{K}^n$ via an application φ. Hence, any application A defined over $\mathbb{L}$ can be seen as an application over $\mathbb{K}^n$ and conversely (just consider $\varphi^{-1} \circ A \circ \varphi$). Recall that any application over $\mathbb{L}$ is a polynomial of $\mathbb{L}[X]$.

The Frobenius Map. The application $\mathrm{F} : X \mapsto X^q$ over $\mathbb{L}$ is called the Frobenius map. It is an automorphism of $\mathbb{L}$ that fixes any element of $\mathbb{K}$. As a consequence, F can also be seen as a matrix $\mathrm{F} \in \mathrm{GL}_n(\mathbb{K})$. A polynomial $P \in \mathbb{L}[X]$ commutes with F if and only if its coefficients are in $\mathbb{K}$.

Linear Polynomials. Let M be an endomorphism of $\mathbb{K}^n$. It can be represented by a matrix over $\mathbb{K}^n$, but also as a polynomial over $\mathbb{L}$. Such $\mathbb{K}$-linear (or "additive") polynomials only have monomials of degree q^i, for $0 \leq i \leq n-1$. In the sequel, we will always identify a $n \times n$ matrix over $\mathbb{K}$ with its polynomial representation over $\mathbb{L}$. The set of matrices commuting with F over $\mathcal{M}_n(\mathbb{K})$ is the $\mathbb{K}$-vector space of dimension n generated by $\left(\mathrm{F}^0, \mathrm{F}, \ldots, \mathrm{F}^{n-1}\right)$. We will also need the following lemma:

Lemma 1. *Let $M \in \mathrm{GL}_n(\mathbb{K})$ be an invertible matrix. If its polynomial representation has coefficients over $\mathbb{K}$, then it is also the case for its inverse.*

Proof. If the polynomial representation of M has coefficients in $\mathbb{K}$, then M commutes with F. This implies that M^{-1} also commutes with F, which in turn implies that the polynomial representation of M^{-1} has coefficients in $\mathbb{K}$. □

2.2 Hidden Field Equations

The HFE scheme was designed in [28] by Patarin. Notice that specific variations of HFE do exist, but we will focus on the basic HFE scheme. Let us briefly recall its mechanism.

Let $\mathbb{K} = \mathbb{F}_q$ be the field with q elements. The HFE secret key is made up of an extension $\mathbb{L}$ of degree n over $\mathbb{K}$, a low-degree polynomial $\mathbf{f}$ over $\mathbb{L}$, and two invertible affine mappings S and T over $\mathbb{K}^n$. The secret polynomial $\mathbf{f}$ has the following particular shape:

$$\mathbf{f}(X) = \sum_{\substack{0 \le i,j \le n \\ q^i+q^j \le d}} a_{i,j} X^{q^i+q^j} + \sum_{\substack{0 \le k \le n \\ q^k \le d}} b_k X^{q^k} + c, \tag{1}$$

with the $a_{i,j}$, the b_k and c lying in $\mathbb{L}$. Polynomials with the same shape as $\mathbf{f}$ are called HFE polynomials[2]. Because decryption requires to invert $\mathbf{f}$, the maximum degree of $\mathbf{f}$, denoted by d, has to be chosen so that the factorization of $\mathbf{f}$ over $\mathbb{L}$ is efficient. All known algorithms for factorizing over finite fields are at least quadratic in the degree of the polynomial, which restricts d to values smaller than about 2^{16}. It also makes sense to consider degree bounds of the form $d = 2 \cdot q^D$, because in equation (1), we may then consider the sum over values of i and j smaller than D. Because the iterates of the Frobenius are $\mathbb{K}$-linear, then $\mathbf{f}$, seen as a transformation of $\mathbb{K}^n$, can be represented represented by a vector of n *quadratic* polynomials in n variables over $\mathbb{K}$. This property extends to the public key of the basic HFE scheme, defined by $\mathbf{PK} = T \circ \circ S$.

2.3 Known Algorithms for Finding Isomorphisms of Polynomials

In this section we briefly list the known techniques to solve the Isomorphism of Polynomials (IP) problem. This problem was first introduced in [28], and its hardness underlies for instance the hardness of the key-recovery of the C^* scheme. As already mentionned, the security of HFE does not rely in general on the hardness of this problem, but in the case of the attack on specific instances presented in this paper, we reduce the recovery of the private key to solving an instance of the IP problem, which happens to be tractable in some cases (e.g. the "subfield" case, see section 5).

Recall that finding a polynomial isomorphism between two vectors of multivariate polynomials $\mathbf{a}$ and $\mathbf{b}$ means finding two invertible matrices U and V in $\mathrm{GL}_n(\mathbb{F}_q)$, as well as two vectors c and d in ${\mathbb{F}_q}^n$ such that:

$$\mathbf{b}\,(x) = V\left(\mathbf{a}\,(U \cdot x + c)\right) + d \tag{2}$$

It has been proved that the IP problem is not NP-hard, unless the polynomial hierarchy collapses [17]. On the other hand, IP has been shown to be as hard as Graph-Isomorphism [33], for which no polynomial algorithms are known.

The first non-trivial algorithm for IP, known as the "To and Fro" technique, is due to Courtois *et al.* [33]. In its primitive form, this algorithm assumes the ability to inverse the polynomial systems, and has therefore an exponential complexity. A theoretical, birthday-based version of this algorithm is claimed to solve the problem in time and space $\mathcal{O}\left(q^{n/2}\right)$ if $c = d = 0$.

In [17], Faugère and Perret present a new technique for solving IP when $c = d = 0$. The idea is to model the problem as an algebraic system of equations and solve it by means of Gröbner bases [5, 8]. This technique has the advantage over the previous one that it is deterministic and always succeeds. On the down side, its complexity is hard to predict. In practice, it turns out to be efficient for instances of IP where the coefficients of all the monomials of all degree of $\mathbf{a}$ and $\mathbf{b}$ are randomly chosen in $\mathbb{F}_q$. For random instances of IP, the practical complexity of [17] has empirically been observed to be $\mathcal{O}\left(n^9\right)$.

[2] They were also studied much earlier in a completely different context by Dembowski and Ostrom [9], so they are sometimes referred to as D–O polynomials in the literature.

More recently, a faster algorithm dealing with the same class of instances ($c = d = 0$) provably achieves an expected complexity of $\mathcal{O}\left(n^6\right)$ on random instances [3]. This means that solving such random instances is feasible in practice for $n = 128$ or $n = 256$, which are the highest values encountered in practical HFE settings.

No polynomial algorithm is known when $c \neq 0$ or $d \neq 0$, or when $\mathbf{a}$ and $\mathbf{b}$ are homogeneous, and these are the most recurring settings in multivariate cryptography. However, it was also shown in [3] that it is possible to solve these hard instances without first guessing c and d. This enables a birthday-based algorithm to deal in practice with these hard instances in time $n^{3.5} \cdot q^{n/2}$.

3 A Specific Family of HFE Secret Polynomials

3.1 A commutation property for some HFE Secret Polynomials

To begin with, let us consider the *à la* C^* case, where the secret polynomial $\mathbf{f}$ over $\mathbb{L}$ is just a monomial $a \cdot X^{q^i+q^j}$, $a \in \mathbb{L}$. Then the public key $\mathbf{PK} = T \circ \mathbf{f} \circ S$ can also be written as $T' \circ X^{q^i+q^j} \circ S$, by "absorbing" the multiplication by the constant a into T. As a consequence, without loss of generality, we can suppose that $a \in \mathbb{K}$ (or even that $a = 1$, but $a \in \mathbb{K}$ suffices for our purpose).

This secret monomial has some special commutation properties, which were used in [12, 13] to perform attacks on SFLASH. More precisely, composing it on the right hand size by multiplications M_x by an element x is equivalent to composing it on the left hand size by $M_{x^{q^i+q^j}}$. Another property, not used in [12, 13], is that it also commutes with the Frobenius map F and its iterates.

When we consider a more general HFE secret polynomial, the two commutation properties no longer hold. However, if we restrict the HFE polynomials to have their coefficients in $\mathbb{K}$, we lose the first property but the commutation with the Frobenius map still remains. Such instances can be represented by figure 1. Notice that if the coefficients of the HFE secret polynomial $\mathbf{f}$ can all be written as the product of the same element u of $\mathbb{L}$ with an element of $\mathbb{K}$, then by considering an equivalent transformation T made up of the original T and the multiplication by this factor u, we can suppose that $\mathbf{f}$ has coefficients in $\mathbb{K}$ too. This is the same as for the monomial case explained above. The same goes if we can modify the transformation S by composing it with a multiplication by an element m over $\mathbb{L}$, in such a way that the remaining polynomial $\mathbf{f}$ has coefficients in $\mathbb{K}$. In fact, this is all about equivalent secret keys [39]. Finally, the commutation property with the Frobenius can be exploited for instances of the type:

$$\mathbf{f}(X) = \sum_{\substack{0 \leq i,j \leq n \\ q^i+q^j \leq d}} u \cdot a_{i,j} \cdot m^{q^i+q^j} \cdot X^{q^i+q^j} + \sum_{\substack{0 \leq k \leq n \\ q^k \leq d}} u \cdot b_k \cdot m^{q^k} \cdot X^{q^k} + u \cdot c, \tag{3}$$

with $a_{i,j}, b_k, c \in \mathbb{K}$, $u \in \mathbb{L}$, and $m \in \mathbb{L}$.

$$\begin{tikzcd}
\mathbb{K}^n \arrow[r, "S"] \arrow[rrr, bend left, "\mathbf{PK}"] & \mathbb{K}^n \arrow[r, dashed, "\mathbf{f}"] & \mathbb{K}^n \arrow[r, "T"] & \mathbb{K}^n
\end{tikzcd}$$

Fig. 1. $\mathbf{PK} = T \circ \mathbf{f} \circ S$. The broken arrow indicates that $\mathbf{f}$ has coefficients in $\mathbb{K}$.

Our key-recovery attack described in Section 4 exploits this second commutation property and could also apply to monomial instances of HFE, but this is not the point of this paper, as it has already been efficiently done [12, 13, 18]. Notice that legitimate users could easily check whether the internal polynomial of their secret keys belongs to the family verifying this commutation property, but we do not detail this fact here. In the next subsection, we give bounds on the number of HFE secret polynomials belonging to the family described.

3.2 An Estimation of the Cardinal of this Family

Let us study the cardinal of the family highlighted in section 3.1. Recall that we consider HFE polynomials with coefficients in $\mathbb{K}$, but also polynomials that can be written $M_\delta \circ \mathbf{f}' \circ M_\lambda$, where $\mathbf{f}'$ has coefficients in $\mathbb{K}$, M_δ (respectively M_λ) is the multiplication by $\delta \in \mathbb{L} \setminus \mathbb{K}$ (respectively $\lambda \in \mathbb{L} \setminus \mathbb{K}$).

Amongst this set of polynomials defined by $M_\delta \circ \mathbf{f}' \circ M_\lambda$, there are some instances for which $\mathbf{f}'$ commutes with multiplication applications. We already mentionned the case where $\mathbf{f}'$ is a monomial, but actually, such a commutative property may arise when $\mathbf{f}'$ is made up of two terms or sometimes more. Let us detail this point. We can write:

$$\begin{aligned}\mathbf{f}' \circ M_\lambda(X) &= \sum_{i,j} a_{i,j} \cdot (\lambda \cdot X)^{q^i+q^j} + \sum_i b_i \cdot (\lambda \cdot X)^{q^i} + c \\ &= \sum_{i,j} a_{i,j} \cdot \lambda^{q^i+q^j} \cdot X^{q^i+q^j} + \sum_i b_i \cdot \lambda^{q^i} \cdot X^{q^i} + c; \qquad (4) \\ M_\delta \circ \mathbf{f}'(X) &= \sum_{i,j} a_{i,j} \cdot \delta \cdot X^{q^i+q^j} + \sum_i b_i \cdot \delta \cdot X^{q^i} + \delta \cdot c. \qquad (5)\end{aligned}$$

When $\mathbb{K} = \mathbb{F}_2$ and $c = 0$, or $\mathbb{K} \neq \mathbb{F}_2$ and $\mathbf{f}'$ is homogeneous, we can sometimes have an equality between the two right-hand sides of equations (4) and (5) above. Let us consider these instances and suppose we have such an equality. As a result, we have some conditions on δ and λ. More precisely, we see that for a commutation property with multiplications to exist, the conditions, when consistents, often force λ and δ to live in a strict subfield of $\mathbb{L}$, which turns out to be most probably $\mathbb{K}$ as soon as $\mathbf{f}'$ has more than two terms. These are very specific instances, but have to be considered if we want to evaluate the number of HFE secret keys which belong to the family described in subsection 3.1. We have:

Proposition 2. *The number of HFE secret polynomials belonging to the family considered in this paper is lower-bounded by:*

i) $(q^{\frac{(D+1)(D+2)}{2}} - 1) \cdot q^{D+2} \cdot (q^n - q)$, when $\mathbb{K} \neq \mathbb{F}_2$,
ii) $(q^{\frac{D(D+1)}{2}} - 1) \cdot q^{D+3} \cdot (q^n - q)$, when $\mathbb{K} = \mathbb{F}_2$,

corresponding to $\mathcal{O}\left(q^{D^2+n}\right)$*, and upper-bounded by:*

i) $\left((q^{\frac{(D+1)(D+2)}{2}} - 1) \cdot q^{D+2} - \frac{(D+1)(D+2)}{2} \cdot (q-1)\right) \cdot (q^n - q)^2 + \frac{(D+1)(D+2)}{2} \cdot (q-1) \cdot (q^n - q)$, when $\mathbb{K} \neq \mathbb{F}_2$,
ii) $\left((q^{\frac{D(D+1)}{2}} - 1) \cdot q^{D+2} - \frac{(D+1)(D+2)}{2} \cdot (q-1)\right) \cdot (q^n - q)^2 + \frac{(D+1)(D+2)}{2} \cdot (q-1) \cdot (q^n - q)$, when $\mathbb{K} = \mathbb{F}_2$,

which corresponds to $\mathcal{O}\left(q^{D^2+2n}\right)$.

Proof. An HFE polynomial has $\frac{(D+1)(D+2)}{2} + (D+1) + 1 = \frac{D(D+5)}{2} + 3$ terms when $\mathbb{K} \neq \mathbb{F}_2$, $\frac{(D+1)(D+2)}{2} - (D+1) + (D+2) + 1 = \frac{(D+1)(D+2)}{2} + 2$ when $\mathbb{K} = \mathbb{F}_2$. We have:

1. $\mathbb{K}$ has q elements. The number of HFE polynomials with coefficients in $\mathbb{K} \neq \mathbb{F}_2$ is $q^{\frac{D(D+5)}{2}+3}$ ($q^{\frac{(D+1)(D+2)}{2}+2}$ when $\mathbb{K} = \mathbb{F}_2$). We however focus on polynomials over $\mathbb{K}$, which are non-linear over $\mathbb{K}$. This gives $(q^{\frac{(D+1)(D+2)}{2}} - 1) \cdot q^{D+2}$ polynomials when $\mathbb{K} \neq \mathbb{F}_2$, $(q^{\frac{(D+1)(D+2)}{2}-(D+1)} - 1) \cdot q^{D+3} = (q^{\frac{D(D+1)}{2}} - 1) \cdot q^{D+3}$ otherwise.
2. The number of elements belonging to $\mathbb{L} \setminus \mathbb{K}$ is $q^n - q$. Hence, the number of HFE polynomials that can be written as a polynomials with coefficients in $\mathbb{K}$ (a polynomial of point 1.), composed on the left by a mulitplication M_λ, for $\lambda \in \mathbb{L} \setminus \mathbb{K}$, is:
$$\begin{aligned} &(q^{\frac{(D+1)(D+2)}{2}} - 1) \cdot q^{D+2} \cdot (q^n - q), && \text{when } \mathbb{K} \neq \mathbb{F}_2, \\ &(q^{\frac{D(D+1)}{2}} - 1) \cdot q^{D+3} \cdot (q^n - q), && \text{when } \mathbb{K} = \mathbb{F}_2. \end{aligned}$$
This corresponds to the lower bound of the proposition.

Now, to evaluate the exact number of polynomials belonging to our family, we should evaluate the number of polynomials that can be written as in point 2, composed on the right by a multiplication by an element of $\mathbb{L} \setminus \mathbb{K}$. However, we saw that some polynomials have the property that composing them by a multiplication on the left is equivalent to composing them by another multiplication on the right. We thus have to be carefull not to count such poynomial twice. Amongst these polynomials, only monomials have this property for sure. The number of $\mathbb{K}$-quadratic monomials over $\mathbb{L}$ with coefficients in $\mathbb{K}$ is $\frac{(D+1)(D+2)}{2} \cdot (q-1)$ or $\frac{(D(D+1)}{2}) \cdot (q-1)$, whether $\mathbb{K} = \mathbb{F}_2$ or not. This allows to establish the upper-bound of the proposition:

$$\begin{aligned} &\left((q^{\frac{(D+1)(D+2)}{2}} - 1) \cdot q^{D+2} - \frac{(D+1)(D+2)}{2} \cdot (q-1) \right) \cdot (q^n - q)^2 \\ &+ \frac{(D+1)(D+2)}{2} \cdot (q-1) \cdot (q^n - q), && \text{when } \mathbb{K} \neq \mathbb{F}_2, \\ &\left((q^{\frac{D(D+1)}{2}} - 1) \cdot q^{D+2} - \frac{(D+1)(D+2)}{2} \cdot (q-1) \right) \cdot (q^n - q)^2 \\ &+ \frac{(D+1)(D+2)}{2} \cdot (q-1) \cdot (q^n - q), && \text{when } \mathbb{K} = \mathbb{F}_2. \end{aligned}$$

□

We show in this paper that for all HFE secret polynomials with coefficients in $\mathbb{K}$ (or more precisely, the family described by equation (3) and number in proposition 2), the security in fact relies on the hardness of the IP problem. Moreover, in the cases where this IP problem can be solved (see subsection 2.3), then we can also recover an efficient secret key (maybe different from the original one) in practical time.

4 The Attack

The attack being quite complex, let us give an overview. A pseudo-code of the attack is given in fig. 2. First, we show that the representation of $\mathbb{L}$ can be supposed public. Then, as already mentioned in Section 3.1, we use the commutation of the Frobenius map with the secret polynomials considered, which propagates to the public key **PK**. This key property allows us to recover applications closely related to S and T. An interpolation of **PK** combined with these applications then gives us a polynomial over $\mathbb{K}$ from which we recover $\mathbf{f}$ or an equivalent low-degree polynomial by computing a functional decomposition. In any case, we obtain the original secret key or a different one that allows us to decrypt as efficiently as the secret key owner. All these assertions are detailed and justified in this section.

Fig. 2 Pseudo-code of the attack

```
Require: An HFE public key $\mathbf{PK}$, generated by $(T, \mathbf{f}, S)$ such that $\mathbf{f} \in \mathbb{K}[X]$.
Ensure: An equivalent secret key: $(T', \mathbf{f}', S')$, with $\deg \mathbf{f}' \leq \deg \mathbf{f}$.
// section 4.2
repeat
    Let $(U, V)$ be a (random) solution to the IP problem: $U \circ \mathbf{PK} = \mathbf{PK} \circ V$.
until $U$ and F are similar
// section 4.3
for all $i_0$ in $[1; n-1]$ prime with $n$ do
    Let $k = {i_0}^{-1} \mod n$.
    Compute $\widetilde{S}, \widetilde{T}$ such that $\mathrm{F} = \widetilde{S} \circ V^k \circ \widetilde{S}^{-1} = \widetilde{T}^{-1} \circ U^k \circ \widetilde{T}$.
    // section 4.4
    Interpolate $\mathbf{g} = \widetilde{T}^{-1} \circ \mathbf{PK} \circ \widetilde{S}^{-1}$.
    if $\mathbf{g}$ has coefficients in $\mathbb{K}$ then
        // section 4.5
        Compute $F_1, F_2$ and $\mathbf{f}_2$, such that $\mathbf{g} \circ F_1 = F_2^{-1} \circ \mathbf{f}_2$.
        return $\left(\widetilde{T} \cdot F_2^{-1}, \mathbf{f}_2, F_1^{-1} \cdot \widetilde{S}\right)$
    end if
end for
```

4.1 Equivalent Secret Keys : Irrelevance of Hiding the Extension

It is shown in [39] that there are many equivalent keys in HFE. As a consequence, one can assume S and T to be *linear* bijections (as opposed to affine), and arbitrarily choose their value on one point. Indeed, it is possible to compensate changes in S and T by changes in $\mathbf{f}$. In the restricted setting where $\mathbf{f}$ is assumed to have coefficients in the base field $\mathbb{K}$ (instead of the extension field $\mathbb{L}$), this is no longer possible, because there is not enough freedom if we want to keep $\mathbf{f} \in \mathbb{K}[X]$.

However, what holds in general and still holds for our family of secret keys is that the assumption of keeping the representation φ of $\mathbb{L}$ secret is not necessary. This was already mentioned in the original paper [28], and as a matter of fact, the specifications of both Quartz [31] and SFLASH make the description of the extension public. In any case, it is possible to generate the *same* public key from the *same* secret polynomial, while fixing an arbitrary correspondence between $\mathbb{L}$ and $\mathbb{K}^n$. It simply requires slight modifications on S and T:

Proposition 3. *Let* $\mathbf{SK} = (T, \mathbf{f}, S, \varphi)$ *be an HFE secret key. Then for any choice of an isomorphism* φ' *between* $\mathbb{L}$ *and* $\mathbb{K}^n$*, there exist two affine bijections* S' *and* T' *such that* $\mathbf{SK}' = (T', \mathbf{f}, S', \varphi')$ *is equivalent to* $\mathbf{SK}$ *(i.e., generates the same public key).*

Proof. If φ' denotes another isomorphism between $\mathbb{L}$ and $\mathbb{K}^n$, then $\phi = \varphi' \circ \varphi^{-1}$ is a $\mathbb{K}$-linear invertible application such that $\varphi' = \phi \circ \varphi$. Using the correspondence φ', the composition $T' \circ \mathbf{f} \circ S'$ is also equal to $\mathbf{PK}$, where $T' = \phi \circ T$ and $S' = S \circ \phi^{-1}$. □

Thus, the assumption of keeping $\mathbb{L}$ secret does not have an influence on the security of HFE. Would the extension be secret, one could just arbitrarily fix its own and be guaranteed that an equivalent secret key exists. As a consequence, throughout the sequel, we assume that the description of $\mathbb{L}$ is public.

4.2 A Useful Property of HFE Secret Polynomials Lying in $\mathbb{K}[X]$

Recall from Section 2.1 that because $\mathbf{f}$ has coefficients in $\mathbb{K}$, then it commutes with F:

$$\mathbf{f} \circ \mathrm{F}(X) = \mathrm{F} \circ \mathbf{f}(X) \tag{6}$$

Patarin left as an open problem whether this property has security implications or not. We shall demonstrate that it does indeed. Most importantly, this property is detectable on the public-key.

Proposition 4. *There exists non-trivial polynomial isomorphisms between the public key and itself. More precisely, the invertible mapping ψ defined below transforms a matrix M that commutes with $\mathbf{f}$ into a solution of the polynomial automorphism of the public-key:*

$$\psi : M \mapsto \left(T \cdot M^{-1} \cdot T^{-1}, S^{-1} \cdot M \cdot S\right)$$

As a consequence, $\psi(F), \ldots, \psi\left(F^{n-1}\right)$ are non-trivial isomorphisms between $\mathbf{PK}$ *and itself.*

Proof. Let M be a matrix such that $\mathbf{f} \circ M = M \circ \mathbf{f}$. Then we get:

$$\begin{aligned}\mathbf{PK} \circ (S^{-1} \circ M \circ S) &= T \circ \mathbf{f} \circ S \circ S^{-1} \circ M \circ S \\ &= T \circ M \circ \mathbf{f} \circ S \\ &= (T \circ M \circ T^{-1}) \circ \mathbf{PK} \\ \Leftrightarrow \mathbf{PK} &= (T \circ M \circ T^{-1})^{-1} \circ \mathbf{PK} \circ (S^{-1} \circ M \circ S)\end{aligned}$$

Then, because of (6), $\psi(F), \ldots, \psi\left(F^{n-1}\right)$ are automorphisms of the public key. □

Remark 1. The existence of other solutions besides those mentioned in proposition 4 is extremely unlikely. Indeed, this would imply the existence of other linear applications commuting with the (non-linear) internal polynomial. However, besides the monomial instances, where multiplication matrices commute in some sense with $\mathbf{f}$, we are not aware of instances that would verify such a property. Thus, if we consider a particular solution of the problem of retrieving an automorphism of the public-key, we can assume that it is $\psi\left(F^{i_0}\right)$, for some unknown power i_0.

Hardness of the IP Problem. We discussed algorithms for solving the IP problem in subsection 2.3. In our setting, the conditions for which the polynomial-time IP algorithms are applicable are:

i) The secret transformations S and T are linear (as opposed to affine).
ii) The b_k coefficients of (1) are not all zero.
iii) The c coefficient of (1) is non-zero.

The first condition can only be satisfied if choosing linear S and T was a deliberate decision (otherwise it will only happen with negligible probability). There are good reasons of doing so: first it reduces a bit the size of the private key. Second, in general, because of the existence of equivalent keys, it can be assumed that S and T are linear. However, we emphasize that this last fact is *no longer true* if the internal polynomial $\mathbf{f}$ is chosen in $\mathbb{K}[X]$ instead of $\mathbb{L}[X]$ (section 4.1).

A sequence of bad design decisions could still lead to the combination of a restricted $\mathbf{f}$ *and* linear S and T.

The second condition will always be satisfied with high probability, and the third will be satisfied with probability $1/q$. It must be noted that if $c = 0$ in (1), then the public-key sends zero to zero, which might not be desirable.

In the case where S and T are affine, the situation is much more painful, and breaking the IP instance in practice requires a workload of $q'^{n/2}$ if the coefficients of S live in $\mathbb{F}_{q'}$. In the case of the "subfield variant" though, since $q' = 2$ and n is small enough, breaking the IP instances is still tractable (see section 5).

4.3 Retrieving "nearly S" and "nearly T" Applications

Let us assume that we have found an automorphism $(U, V) = \psi\left(\mathrm{F}^{i_0}\right)$ of the public-key, for some unknown integer i_0 in the interval $[1; n-1]$. The whole point of the attack is to "extract" enough information about S and T from this automorphism. For this purpose, the value of i_0 has to be known, and it is required that i_0 and n be relatively prime. This latter condition can be easily checked for: F^i and F^j are similar if and only if $\gcd(i, n) = \gcd(j, n)$. Therefore, i_0 is relatively prime with n if U and F are similar. If it turns out not to be the case, we take an other automorphism of $\mathbf{PK}$, until it passes the test. Since there are $\phi(n)$ values of i_0 that are prime with n, we expect to check $n/\phi(n) = \mathcal{O}(\log\log n)$ candidates.

To find out the actual value of i_0, we simply guess its value, and check whether the remaining steps of the attack are carried out successfully. Fortunately, there is a way to discard bad guesses systematically before the most computationally expensive step of the attack, as we will explain in section 4.4.

With the preceding notations, we have the following result:

Proposition 5. *Let* $(U, V) = \psi\left(\mathrm{F}^{i_0}\right)$*, with* $\gcd(i_0, n) = 1$*. Let* k *be such that* $k \cdot i_0 = 1 \bmod n$*.*

i) There exist $\widetilde{S}$, $\widetilde{T}$ *in* $\mathrm{GL}_n(\mathbb{K})$ *such that* $\mathrm{F} = \widetilde{S} \circ V^k \circ \widetilde{S}^{-1}$ *and* $\mathrm{F} = \widetilde{T}^{-1} \circ U^k \circ \widetilde{T}$.

ii) Both $\widetilde{S} \cdot S^{-1}$ *and* $\widetilde{T} \cdot T^{-1}$ *commute with* F.

Proof. *i)* We know that U and V are both similar to F^{i_0}. Thus U^k and V^k are both similar to $\mathrm{F}^{i_0 \cdot k} = \mathrm{F}^{1 \bmod n} = \mathrm{F}$.

ii) Let us consider the case of $\widetilde{S}$ (something similar holds for $\widetilde{T}$). We have:

$$\begin{aligned} \mathrm{F} &= \widetilde{S} \circ V^k \circ \widetilde{S}^{-1} \\ &= \widetilde{S} \circ S^{-1} \circ \mathrm{F}^{i_0 \cdot k} \circ S \circ \widetilde{S}^{-1} \\ &= \widetilde{S} \circ S^{-1} \circ \mathrm{F} \circ S \circ \widetilde{S}^{-1} \end{aligned}$$

And thus $\mathrm{F} \circ \widetilde{S} \circ S^{-1} = \widetilde{S} \circ S^{-1} \circ \mathrm{F}$. □

In practice, $\widetilde{S}$ and $\widetilde{T}$ can be found very efficiently through linear algebra, given that i_0 is known. Note that for now, this proposition cannot be used to test whether our current guess for i_0 is correct, since we do not know S.

4.4 Building an Equivalent Secret Key

The information about S (resp. T) contained in $\widetilde{S}$ (resp. $\widetilde{T}$) can be used to cancel the action of S and T on the public key. It follows from proposition 5 (see also section 2) that $F_1 = \widetilde{S} \cdot S^{-1}$ and $F_2 = T^{-1} \cdot \widetilde{T}$ are linear combinations over $\mathbb{K}$ of powers of F. We immediately obtain that:

$$\begin{aligned} \widetilde{T}^{-1} \circ \mathbf{PK} \circ \widetilde{S}^{-1} &= F_2^{-1} \circ T^{-1} \circ T \circ \mathbf{f} \circ S \circ S^{-1} \circ F_1^{-1} \\ &= F_2^{-1} \circ \mathbf{f} \circ F_1^{-1}. \end{aligned} \tag{7}$$

We therefore define:

$$\mathbf{g} = \widetilde{T}^{-1} \circ \mathbf{PK} \circ \widetilde{S}^{-1} \bmod \left(X^{q^n} - X \right)$$

Because the HFE polynomials are stable by left and right composition by additive polynomials and by reduction modulo $X^{q^n} - X$, the "peeled off" polynomial $\mathbf{g}$ is still an HFE polynomial. Thus $\mathbf{g}$ has $\mathcal{O}\left(n^2\right)$ coefficients, and that they can be uniquely determined in polynomial time by interpolation (this was noted in [22]. Note that there would not be a unique solution if we did not perform the modular reduction of $\mathbf{g}$). By doing so, we obtain an equivalent secret key, namely $\left(\widetilde{T}, \mathbf{g}, \widetilde{S}\right)$.

By itself, this equivalent key is not particularly useful, since the degree of $\mathbf{g}$ is typically q^n, and we are therefore still facing our initial task of factorizing a sparse polynomial of very high degree. However, $\mathbf{g}$ has a very important property which brings us one step closer to the original secret-key:

Proposition 6. *The coefficients of* $\mathbf{g}$ *are in* $\mathbb{K}$ *(and not in* $\mathbb{L}$*).*

Proof. By hypothesis, the coefficients of $\mathbf{f}$ are in $\mathbb{K}$. From proposition 5, we have that the coefficients of the polynomial representation of F_1 and F_2 are in $\mathbb{K}$, then, so are those of the polynomial representations of ${F_1}^{-1}$ and ${F_2}^{-1}$ (by lemma 1). □

The result of proposition 6 is illustrated in figure 3. This figure also helps remembering how the applications introduced so far intervene.

$$\begin{tikzcd}
\mathbb{K}^n \arrow[r, "S"] \arrow[rrr, bend left, "\mathbf{PK}"] \arrow[dr, "\widetilde{S}"'] & \mathbb{K}^n \arrow[r, dashed, "\mathbf{f}"] \arrow[d, dashed, "F_1"'] & \mathbb{K}^n \arrow[r, "T"] & \mathbb{K}^n \\
& \mathbb{K}^n \arrow[r, dashed, "\mathbf{g}"] & \mathbb{K}^n \arrow[u, dashed, "F_2"'] \arrow[ur, "\widetilde{T}"'] &
\end{tikzcd}$$

Fig. 3: $\mathbf{PK} = T \circ \mathbf{f} \circ S = \widetilde{T} \circ \mathbf{g} \circ \widetilde{S}$. Broken arrows stand for applications with coefficients in $\mathbb{K}$.

This proposition can be used to verify if our guess for i_0 was right. Indeed, if $\mathbf{g}$ is found not to be in $\mathbb{K}[X]$, then the guess was wrong. We are aware that the fact that $\mathbf{g} \in \mathbb{K}[X]$ does not rigorously prove that we have found the right value of i_0. However, it does not matter, as $\mathbf{g} \in \mathbb{K}[X]$ is sufficient for the subsequent step to work.

4.5 Recovering a Low-Degree Equivalent Secret Key

To be useful, an equivalent secret key must have an internal polynomial of low degree. We now show how to obtain one, by actually computing the decomposition given by equation (7) of Section 4.4. This is in fact a *much easier* problem than computing the equivalent decomposition on the original public key, because we deal with applications whose coefficients belong to $\mathbb{K}$. They are then left invariant by the Frobenius (hence by F_1 and F_2), which implies that the problem of finding the decomposition reduces to finding a solution of an overdefined system of quadratic equations. This system can be solved in practical time by computing a Gröbner basis, as we now show. To this end, we introduce the following notations:

$$F_1(X) = \sum_{k=0}^{n-1} x_k X^{q^k} \qquad F_1^{-1}(X) = \sum_{k=0}^{n-1} y_k X^{q^k}$$

$$F_2(X) = \sum_{k=0}^{n-1} z_k X^{q^k} \qquad F_2^{-1}(X) = \sum_{k=0}^{n-1} t_k X^{q^k}$$

$$\begin{aligned}
\mathbf{g}(X) &= \sum_{q^i+q^j<q^n} a_{ij} X^{q^i+q^j} + \sum_{i=0}^{n-1} b_i X^{q^i} + c \\
\mathbf{f}_2(X) &= \sum_{q^i+q^j\leq d} e_{ij} X^{q^i+q^j} + \sum_{q^i\leq d} f_i X^{q^i} + g
\end{aligned}$$

Then, we consider the following polynomial equation, also represented by figue 4.5, obtained by composing both sides of equation (7) of Section 4.4 with F_1:

$$\mathbf{g} \circ F_1 = F_2^{-1} \circ \mathbf{f}_2. \tag{8}$$

$$\begin{tikzcd}
\mathbb{K}^n \arrow[r, dotted, "\mathbf{f}"] \arrow[d, dotted, "F_1"'] & \mathbb{K}^n \\
\mathbb{K}^n \arrow[r, "\mathbf{g}"] & \mathbb{K}^n \arrow[u, dotted, "F_2"']
\end{tikzcd}$$

Fig. 4: $\mathbf{g} = F_2^{-1} \circ \mathbf{f} \circ F_1^{-1}$. Broken arrows stand for applications with unknown coefficients.

The left-hand side becomes:

$$\begin{aligned}
\mathbf{g} \circ F_1 &= \sum a_{ij} \left(\sum_{k=0}^{n-1} x_k X^{q^k} \right)^{q^i+q^j} + \sum b_i \left(\sum_{k=0}^{n-1} x_k X^{q^k} \right)^{q^i} + c \\
&= \sum_{i,j,k,l} a_{ij} \cdot x_k \cdot x_l \cdot X^{q^{i+k}+q^{l+j}} + \sum_{i,k} b_i \cdot x_k \cdot X^{q^{i+k}} + c
\end{aligned}$$

We obtain a polynomial whose coefficients are quadratic in the coefficients of F_1. Now, let us compute the right-hand side:

$$\begin{aligned}
F_2^{-1} \circ \mathbf{f}_2 &= \sum_{k=0}^{n-1} t_k \left(\sum_{q^i+q^j\leq d} e_{ij} X^{q^i+q^j} + \sum_{q^i\leq d} f_i X^{q^i} + g \right)^{q^k} \\
&= \sum_{i,j,k} t_k \cdot e_{ij} \cdot X^{q^{i+k}+q^{j+k}} + \sum_{i,k} t_k \cdot f_i \cdot X^{q^{i+k}} + g \cdot \sum_{k} t_k
\end{aligned}$$

We again obtain a polynomial whose coefficients are quadratic in the coefficients of both $\mathbf{f}_2$ and F_2^{-1}. Reducing both sides modulo $X^{q^n} - X$ and identifying coefficient-wise the two sides of equation (8) yields a system of $\mathcal{O}\left(n^2\right)$ quadratic equations in $\mathcal{O}\left(n + D^2\right)$ unknowns. However, these equations admit many parasitic solutions (for example, $F_1 = \mathbf{f}_2 = 0$). To avoid these, we also encode the fact that F_1 and F_2 are invertible. We describe how we encode the invertibility of F_1, as this is similar for F_2. We start from the equation: $F_1 \circ F_1^{-1}(X) = X$: because the coefficients of the left-hand side are quadratic in the x_k's and y_k's, we obtain n quadratic equations by reducing the LHS modulo $X^{q^n} - X$ and equating the coefficients on both sides of the equation. Note that this also introduce in all $2n$ additional unknowns (n for the coefficients of F_1^{-1} and n for the coefficients of F_2).

All in all, assuming that the degree of $\mathbf{f}$ is $d = 2q^D$, this yields $n(n+3)/2 + 1$ equations in $4n + D(D+5)/2 + 4$ variables, not counting eventual field equations (one per variable). The existence of at least one solution is guaranteed, because of equation (7) of Section 4.4, as long as we picked the right power of the Frobenius matrix in section 4.2. In fact, even though we just need one, we know that many solutions exist: for instance because the Frobenius commutes with everything in equation (8), we can take a particular solution, compose both F_2^{-1} and F_1 with the Frobenius, and obtain a new solution.

It turns out that these equations can be solved efficiently, even though the number of variables is higher than what is usually tractable, because it is very overdetermined: we have $\mathcal{O}\left(n^2\right)$ equations in $\mathcal{O}\left(n + D^2\right)$ variables, and D has to be small for decryption to be efficient (*i.e.*, $D = \mathcal{O}(\log n)$). In this setting, computing a Gröbner basis turns out to be feasible in practice.

Conjecture 1. The Gröbner basis of a system of random quadratic equations with the same number of variable and polynomials as our equations can be computed by manipulating polynomials of degree at most 8. Thus, it can be computed in time at most $\mathcal{O}\left(n^{24}\right)$ by the F_4 or F_5 algorithms [14, 15]. This is true if D is fixed, or even if grows polynomially with $\log n$.

Justification of the Conjecture. We argue that the complexity of computing a Gröbner basis of our equations is fact polynomial under realistic assumptions, although in the general case the algorithms involved in the computation are simply or doubly exponential.

The usual strategy to solve such an overdefined system of equations is to compute a Gröbner basis for the graded reverse lexicographic order, since it is easier, and then to convert it to a Gröbner basis for the lexicographic order. Let us recall that the complexity of all known Gröbner bases algorithms depends on the *degree of regularity* of the system [7, 1]. This corresponds to the maximal degree of polynomials manipulated during a Gröbner basis computation. If d_{reg} is the degree of regularity of an ideal $I \subset k[x_1, \ldots, x_m]$, then the complexity of computing a Gröbner basis of I using the F_5 algorithm [15] is upper-bounded by:

$$\mathcal{O}\left(\binom{n + d_{reg}}{d_{reg}}^{\omega}\right) = \mathcal{O}\left(n^{\omega \cdot d_{reg}}\right)$$

where ω is the linear algebra constant (between 2 and 3). In general, it is a difficult problem to know *a priori* the degree of regularity, although lower-bounds were shown in the context of the analysis of the XL algorithm [10].

To upper-bound the complexity of our Gröbner-basis computation, we use an existing approximation of the degree of regularity that applies to *regular* and *semi-regular* system of equations (*i.e.*, in which the equations are "as independent as possible". For a formal definition, see [1]). It is conjectured that the proportion of semi-regular systems goes to 1 when n goes to $+\infty$. Therefore, we will assume that for large n a random system is almost surely semi-regular (which is to some extent a worst-case assumption, as it usually means that our system is not easier to solve than the others). The coefficients of the Hilbert series associated with the ideal generated by a semi-regular sequence coincide with those of the series expansion of the function

$f(z) = \left(1 - z^2\right)^m / (1 - z)^n$, up to the degree of regularity. The degree of regularity is the smallest degree d such that the coefficient of degree d in the series expansion of $f(z)$ is not strictly positive. This property enables an explicit computation of the degree of regularity for given values of m and n.

Furthermore, Bardet *et al.* [1] give asymptotic developments of the expression of the degree of regularity in the case of $\alpha \cdot n$ equations in n variables, where α is a constant greater than 1. While this result is not directly applicable to our case (because we have about αn^2 equations), we use it to derive a heuristic expression of the degree of regularity for systems of $\alpha \cdot n^2$.

When there are $\alpha \cdot n$ semi-regular quadratic equations in n variables, [1] gives:

$$d_{reg} = n \left(\alpha - \frac{1}{2} - \sqrt{\alpha(\alpha - 1)} \right) - \frac{a_1}{2(\alpha(\alpha-1))^{\frac{1}{6}}} n^{\frac{1}{3}} - \left(2 - \frac{2\alpha - 1}{4\sqrt{\alpha(\alpha - 1)}} \right) + \mathcal{O}\left(1/n^{1/3}\right), \quad \text{with } a_1 \approx -2.33811. \tag{9}$$

While we are well-aware that it is not theoretically justified (because equation (9) is established for a constant α), we now set $\alpha = \beta n$, and express d_{reg} as a function of β. This yields

$$d_{reg} = \frac{1}{8\beta} - \frac{a_1}{2\beta^{1/3}} - \frac{3}{2} + \mathcal{O}\left(1/n\right). \tag{10}$$

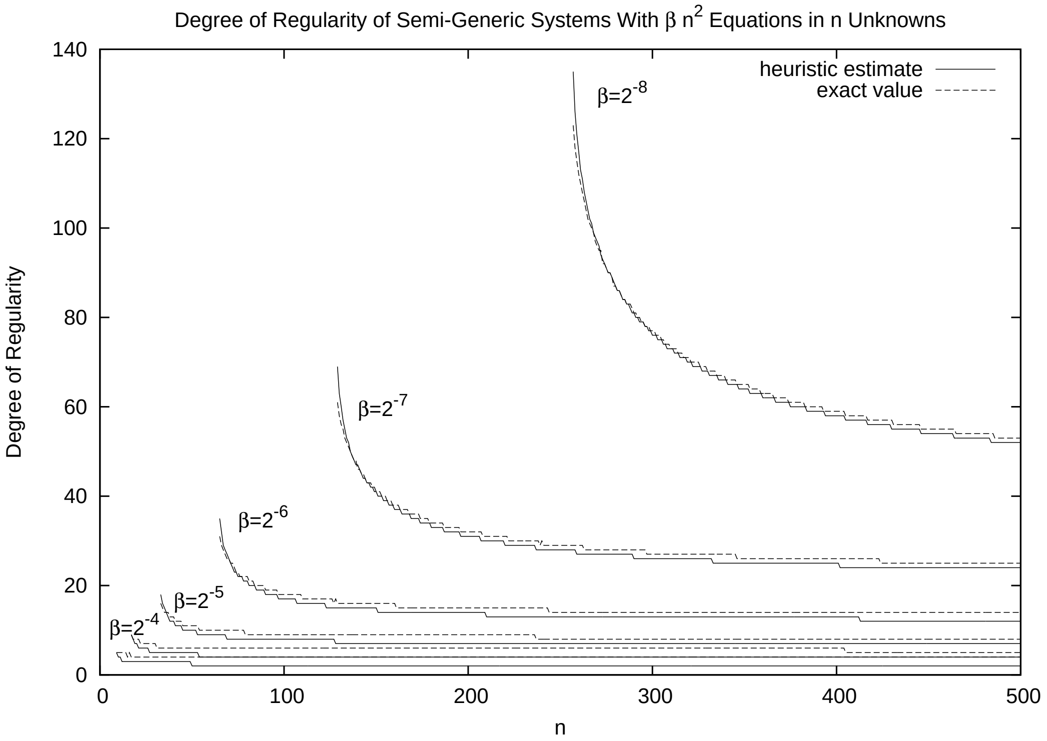
This heuristic result can be empirically checked to be rather precise, for various values of β and n, as shown in fig. 5(a). When n grows to infinity, it seems that the degree of regularity converges to a constant, an approximation of which is given by (10). We now apply this result to our setting:

1. Consider that D is fixed. Then when n becomes big, we have $\beta = 1/32$. Equation (10) then yields $d_{reg} = 7$ for large n (actually computing it using the Hilbert series gives a value of 8 for big n). Computing the Gröbner basis can thus be achieved with complexity $\mathcal{O}\left(n^{8\omega}\right)$.
2. Consider that the degree of $\mathbf{f}$ grows polynomially with n, which means that $D = \mathcal{O}\left(\log n\right)$. In that case we have $\beta = 1/32 + \mathcal{O}\left(\frac{\log n}{n}\right)$, and equation (10) still yields $d_{reg} \approx 7$ for large n.

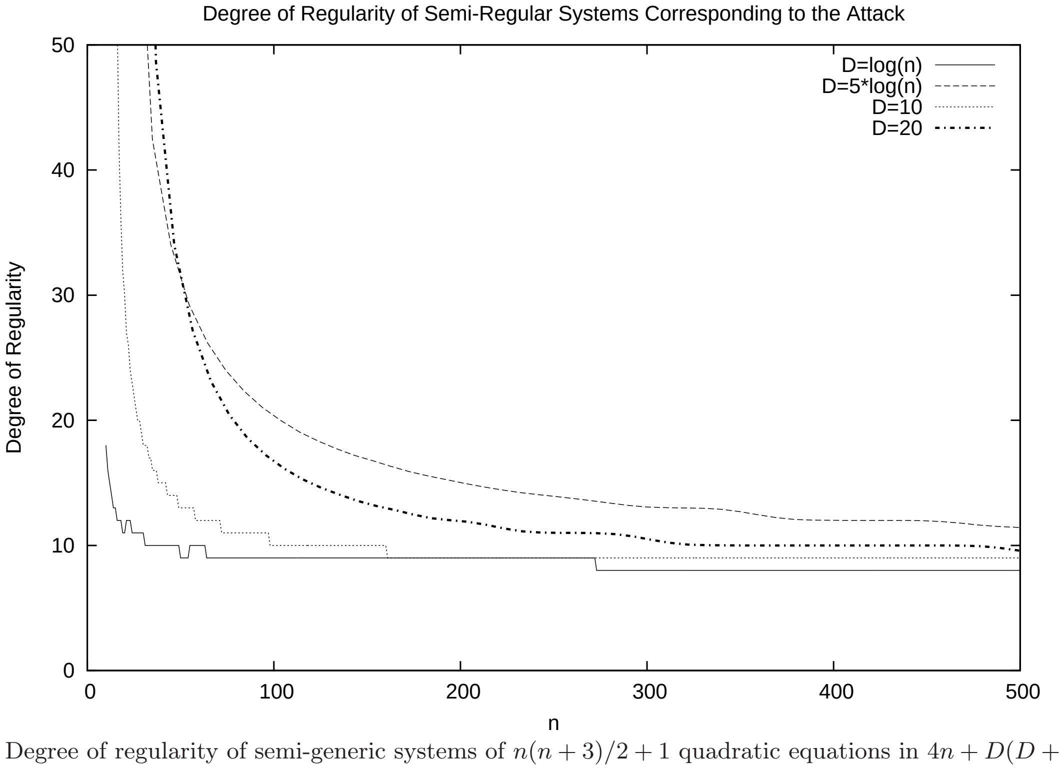
This shows that even in the more general setting the computation of the Gröbner basis should be polynomial, and the degree of the polynomials should not increase beyond a given threshold. Fig. 5(b) shows the degree of regularity of systems having the same parameters as those considered in the attack.

Comments and Practical Results. While the result conjectured above means that computing the polynomial decomposition we are dealing with should be polynomial, some remarks are in order. First, our equations are not random, not to mention semi-regular. This follows from the fact that they admit many solution, while a random overdetermined system has no solutions with overwhelming probability. Next, our experiments (for various values of n and D) indicate that a Gröbner basis can be computed by manipulating polynomials of degree at most 3, leading to an empirical complexity of $\mathcal{O}\left(n^9\right)$. Our equations are thus *easier* to solve than random systems with the same parameters.

Once the equations are solved, we recover an equivalent secret-key $\left(\widetilde{T} \cdot F_2^{-1}, \mathbf{f}_2, F_1^{-1} \cdot \widetilde{S}\right)$, which allows us to decrypt with the same time complexity as the legitimate user, since $\mathbf{f}_2$ has essentially the same degree as $\mathbf{f}$.



(a) Comparison between the heuristic estimate and the actual values of the degree of regularity for $\alpha \cdot n^2$ quadratic equations in n unknowns.



(b) Degree of regularity of semi-generic systems of $n(n+3)/2+1$ quadratic equations in $4n+D(D+5)/2+4$ variables.

5 Applications and Experiments

We programmed the HFE key-generation and encryption, as well as the attack, in the MAGMA [2] computer-algebra system. We do not claim that our implementation is efficient, nor reflects what kind of performances can be obtained in encryption. All the experiments were run on one core of an Intel 2.3Ghz Xeon "Nehalem" computer with 74 Gbyte of RAM. We tested our attack on several sets of parameters described below. We forged the solution of the IP instance from the knowledge of the secret S and T. The actual timings are given in figure 5.

Weak Keys. We first tested the attack on realistically-sized weak keys, corresponding to parameters set A,B and C. The chosen parameters allows the encryption or signature of 256, 134 and 97 bits respectively. We choose the degree of the internal polynomial very conservatively (*i.e.*, much higher than what was proposed for the HFE challenges, and high enough to make decryption painfully slow). To make the IP part of the attack feasible, we choose the secret bijections S and T to be linear (as opposed to affine). Then solving the IP instance is a matter of seconds with the techniques presented in [3]. We emphasize that none of the existing attack can be close to practical on parameter sets A and B.

Patarin's "Subfield" Variant of HFE. In order to reduce the size of the public key, Patarin suggested in [29] a "subfield" variant of HFE, in which the coefficients of the quadratic equations of **PK** live in a subfield $\Bbbk$ of $\mathbb{K}$. If $\mathbb{K} = \mathbb{F}_{256}$ and $k = \mathbb{F}_2$, this reduces the size of the public key by a factor of 8. To achieve this, the coefficients of S and T, the coefficients of the defining polynomial of the extension field $\mathbb{L}$, and the coefficients of the internal polynomial $\mathbf{f}$ have to be chosen in $\Bbbk$ (instead of $\mathbb{K}$ or $\mathbb{L}$ for the latter). S and T will be affine, so the polynomial-time IP algorithms do not apply in this case.

In order for the reduction of the public key size to be effective, $\mathbb{K}$ has to be relatively big and $\Bbbk$ relatively small. The former implies that D cannot be very huge, otherwise decryption is impractical, while the latter means little entropy in the internal polynomial. This opens a possible way of attack, consisting in guessing $\mathbf{f}$ and then solving the IP problem to recover S and T. We shall compare the attack presented in this paper with this simple one.

Patarin's "concrete proposal" is parameter set D in fig. 5. For practical decryption, we have to choose $D = 2$ (yielding an internal polynomial of degree at most 131072), and decryption can take at most 4 minutes on our machine. The internal polynomial has at most 10 terms with coefficients in $\mathbb{F}_2$. The simple "guess-**f**-then-IP" key recovery attack therefore needs to solve 2^{10} affine IP instances for which $q = 2$ and $n = 29$. Such instances are in fact tractable even with older techniques (though no one ever noticed it), for instance using the "to-and-fro" algorithm of [33]. In that case, the "guess-then-IP" attack has a workload of 2^{68}. With the new attack presented in this paper, and the more advanced IP techniques described in [3], solving the IP instance takes about one second, and our attack takes less than one minute.

To show that the "subfield" variant is broken beyond repair, we show that it is possible to attack in practice parameters twice as big as the concrete proposal. This is parameter set E. The internal polynomial now has 21 terms, so the simple attack requires breaking 2^{21} affine instances of the IP problem with $q = 2$ and $n = 59$. According to [3], breaking one of these instance should take about one month using inexpensive hardware, with a workload of about 2^{59}. The "guess-then-IP" attack is here clearly impractical with a complexity of 2^{80}. Our attack requires one month to break the IP instance, plus about 4 hours for the remaining steps.

Parameter set	A	B	C	D	E
block size (bits)	256	134	97	232	236
q	256	4	2	256	16
N	32	67	97	29	59
$\deg \mathbf{f}$	131072	131072	128	131072	131072
coefficients of $\mathbf{f}$ in	$\mathbb{F}_{256}$	$\mathbb{F}_4$	$\mathbb{F}_2$	$\mathbb{F}_2$	$\mathbb{F}_2$
S and T	linear	linear	linear	affine	affine
coefficients of S, T in	$\mathbb{F}_{256}$	$\mathbb{F}_4$	$\mathbb{F}_2$	$\mathbb{F}_2$	$\mathbb{F}_2$
Terms in $\mathbf{f}$	10	54	29	10	21
size of **PK** (bits)	143'616	314'364	461'138	13'485	107'970
IP	polynomial			$\approx$ 1s	$\approx$ 5 weeks
Interpolation of $\mathbf{g}$ (once)	79s	30 min	140 min	51s	23min
Gröbner	7h	1 day	1 week	45s	3h
Variables / Equations	136 / 593	322/4947	423/10028	124 / 494	253 / 1889
Memory required	2.1Gbyte	45Gbyte	180Gbyte	350Mbyte	13.9Gbyte
Order Change	15s	30 min	4h	0s	30s

Fig. 5: Timings for the Attack

6 Conclusion

In this paper, we considered a special family of HFE instances, where the internal secret polynomial is defined over the base field $\mathbb{K}$ instead of the extension field $\mathbb{L}$. We show that, in that case, there are non-trivial isomorphisms of polynomials between the corresponding public key and itself. Interestingly, finding such an isomorphism suffices to completely recover (in practical time) a secret-key that allows fast decryption.

References

1. Bardet, M., Faugère, J.C., Salvy, B., Yang, B.Y.: Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems. In: MEGA'05. (2005) Eighth International Symposium on Effective Methods in Algebraic Geometry, Porto Conte, Alghero, Sardinia (Italy), May 27th – June 1st.
2. Bosma, W., Cannon, J.J., Playoust, C.: The Magma Algebra System I: The User Language. J. Symb. Comput. **24**(3/4) (1997) 235–265
3. Bouillaguet, C., Faugère, J.C., Fouque, P.A., Pérret, L.: Isomorphism of Polynomials : New Results (October 2010) unpublished manuscript. Available at: `http://www.di.ens.fr/~bouillaguet/pub/ip.pdf`.
4. Braeken, A., Wolf, C., Preneel, B.: A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes. In Menezes, A., ed.: CT-RSA. Volume 3376 of Lecture Notes in Computer Science., Springer (2005) 29–43
5. Buchberger, B.: Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal. PhD thesis, University of Innsbruck (1965)
6. Buss, J.F., Frandsen, G.S., Shallit, J.: The Computational Complexity of Some Problems of Linear Algebra. J. Comput. Syst. Sci. **58**(3) (1999) 572–596
7. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
8. Cox, D.A., Little, J.B., O'Shea, D.: Ideals, Varieties and Algorithms. Springer (2005)
9. Dembowski, P., Ostrom, T.G.: Planes of Order n with Collineation Groups of Order n^2. Mathematische Zeitschrift **103**(3) (1968) 239–258

10. Diem, C.: The xl-algorithm and a conjecture from commutative algebra. In Lee, P.J., ed.: ASIACRYPT. Volume 3329 of Lecture Notes in Computer Science., Springer (2004) 323–337
11. Ding, J., Schmidt, D., Werner, F.: Algebraic Attack on HFE Revisited. In Wu, T.C., Lei, C.L., Rijmen, V., Lee, D.T., eds.: ISC. Volume 5222 of Lecture Notes in Computer Science., Springer (2008) 215–227
12. Dubois, V., Fouque, P.A., Shamir, A., Stern, J.: Practical Cryptanalysis of SFLASH. In: CRYPTO. Volume 4622., Springer (2007) 1–12
13. Dubois, V., Fouque, P.A., Stern, J.: Cryptanalysis of SFLASH with Slightly Modified Parameters. In: EUROCRYPT. Volume 4515., Springer (2007) 264–275
14. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra **139**(1-3) (June 1999) 61–88
15. Faugère, J.C.: A New Efficient Algorithm for Computing Gröbner Bases Without Reduction to Zero (F5). In: ISSAC '02: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, New York, NY, USA, ACM (2002) 75–83
16. Faugère, J.C., Joux, A.: Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In Boneh, D., ed.: CRYPTO. Volume 2729 of Lecture Notes in Computer Science., Springer (2003) 44–60
17. Faugère, J.C., Perret, L.: Polynomial Equivalence Problems: Algorithmic and Theoretical Aspects. In Vaudenay, S., ed.: EUROCRYPT. Volume 4004 of Lecture Notes in Computer Science., Springer (2006) 30–47
18. Fouque, P.A., Macario-Rat, G., Stern, J.: Key Recovery on Hidden Monomial Multivariate Schemes. In Smart, N.P., ed.: EUROCRYPT. Volume 4965 of Lecture Notes in Computer Science., Springer (2008) 19–30
19. Gilbert, H., Minier, M.: Cryptanalysis of SFLASH. In Knudsen, L.R., ed.: EUROCRYPT. Volume 2332 of Lecture Notes in Computer Science., Springer (2002) 288–298
20. Granboulan, L., Joux, A., Stern, J.: Inverting HFE Is Quasipolynomial. In Dwork, C., ed.: CRYPTO. Volume 4117 of Lecture Notes in Computer Science., Springer (2006) 345–356
21. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar Signature Schemes. In: EUROCRYPT. (1999) 206–222
22. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. In Wiener, M.J., ed.: CRYPTO. Volume 1666 of Lecture Notes in Computer Science., Springer (1999) 19–30
23. Matsumoto, T., Imai, H.: Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. In: EUROCRYPT. (1988) 419–453
24. McEliece, R.: A Public-Key Cryptosystem Based on Algebraic Coding Theory (1978) DSN Progress Report 42-44.
25. Naccache, D., ed.: Topics in Cryptology - CT-RSA 2001, The Cryptographer's Track at RSA Conference 2001, San Francisco, CA, USA, April 8-12, 2001, Proceedings. In Naccache, D., ed.: CT-RSA. Volume 2020 of Lecture Notes in Computer Science., Springer (2001)
26. Ore, O.: Contributions to The Theory of Finite Fields. Transactions A. M. S. **36** (1934) 243–274
27. Patarin, J.: Cryptoanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88. In Coppersmith, D., ed.: CRYPTO. Volume 963 of Lecture Notes in Computer Science., Springer (1995) 248–261
28. Patarin, J.: Hidden fields equations (hfe) and isomorphisms of polynomials (ip): Two new families of asymmetric algorithms. In: EUROCRYPT. (1996) 33–48
29. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: EUROCRYPT. (1996) 33–48 Etended version available on `http://www.minrank.org/hfe.pdf`.
30. Patarin, J., Courtois, N., Goubin, L.: Flash, a fast multivariate signature algorithm. [25] 298–307
31. Patarin, J., Courtois, N., Goubin, L.: QUARTZ, 128-Bit Long Digital Signatures. [25] 282–297
32. Patarin, J., Goubin, L.: Asymmetric cryptography with s-boxes. In Han, Y., Okamoto, T., Qing, S., eds.: ICICS. Volume 1334 of Lecture Notes in Computer Science., Springer (1997) 369–380
33. Patarin, J., Goubin, L., Courtois, N.: Improved Algorithms for Isomorphisms of Polynomials. In: EUROCRYPT. (1998) 184–200
34. Ritt, J.F.: Prime and Composite Polynomials. American M. S. Trans. **23** (1922) 51–66

35. Sidorenko, A.V., Gabidulin, E.M.: The Weak Keys For HFE. In: 7th International Symposium on Communication Theory and Applications. (2003) 239–244
36. Tao, R.J., Chen, S.H.: Two varieties of finite automaton public key cryptosystem and digital signatures. Journal of computer science and technology **1**(1) (1986) 9–18
37. von zur Gathen, J.: Functional Decomposition of Polynomials: The Tame Case. J. Symb. Comput. **9**(3) (1990) 281–299
38. von zur Gathen, J.: Functional Decomposition of Polynomials: The Wild Case. J. Symb. Comput. **10**(5) (1990) 437–452
39. Wolf, C., Preneel, B.: Large Superfluous Keys in Multivariate Quadratic Asymmetric Systems. In Vaudenay, S., ed.: Public Key Cryptography. Volume 3386 of Lecture Notes in Computer Science., Springer (2005) 275–287