J. Math. Cryptol. **5** (2011), 247–275
DOI 10.1515/JMC.2011.012

A family of weak keys in HFE and the corresponding practical key-recovery

Charles Bouillaguet, Pierre-Alain Fouque, Antoine Joux and Joana Treger

Communicated by Rainer Steinwandt

Abstract. The HFE (hidden field equations) cryptosystem is one of the most interesting public-key multivariate schemes. 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 the hardness of recovering the secret-key of schemes such as SFLASH or C^* 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.

Keywords. Cryptanalysis, multivariate cryptography, HFE, weak keys, Gröbner bases.

2010 Mathematics Subject Classification. 08-04, 12Yxx.

1 Introduction

Multivariate cryptography is interesting from several points of view. First of all, the hard problem it is based on, namely solving systems of multivariate equations, is natural, well-studied and only generic algorithms with exponential worst-case complexity are known to solve it. Multivariate cryptography has been proposed as an alternative to the RSA cryptosystem since the underlying hard problem can-

The first author is supported by a grant from the EADS foundation.

not be attacked faster by quantum computers. Finally, it is appealing since the public operations do not require computations with large integers, and no cryptoprocessor would be needed on smartcards.

In cryptography, it is often preferable to work with multivariate quadratic polynomials for efficiency reasons, and the corresponding problem, finding the common zeroes of a collection of multivariate quadratic polynomials, is called the MQ Problem. It is well known that the MQ Problem is NP-complete over any finite field [19]. Instances of 3-SAT, for instance, can easily be transformed into polynomially-bigger instances of MQ over $\mathbb{F}_2$.

The HFE cryptosystem has been proposed in 1996 by Patarin in [29] in order to avoid his attack on the Matsumoto–Imai cryptosystem [25,28]. The latter 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 $\mathbb{F}_q$, by composing it with 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 [29], 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.

Related work. 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 [23].

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 [36]), and merely deciding the existence of roots is in fact NP-complete [23].

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 [31], FAPKC [37], 2R [33] and McEliece [26]. 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 [27, 35]. Today, polynomial decomposition algorithms exist for some classes of polynomials over finite fields [38, 39], 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. This suggest that d should be an increasing function of n to make the complexity of the attacks superpolynomial. However, in order for decryption to be polynomial, d must grow at most polynomially in n. We consider this setting to be natural, because decryption is polynomial, yet no known polynomial attack exists. We will therefore assume in the remainder of the paper that d is polynomial in n.

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. [21] showed that specific algebraic properties of the HFE equations make the complexity of inverting HFE subexponential, in time $\mathcal{O}(\exp(\log^2 n))$.

In general, the hardness of recovering the secret key of HFE from the public key is unrelated to the Isomorphism of Polynomials (IP) Problem [29], unless the internal polynomial is made public. A key recovery attack in the usual case where this polynomial is secret was presented in [23] 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 $\mathcal{O}(\exp(\log^3 n))$. 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. Furthermore, Dubois and Gama have studied the degree of regularity of various classes of HFE instances in [13]. While they only provide an upper bound, their result shows that there are wide ranges of parameters that are not provably broken by the direct Gröbner attack. Moreover, standard modification – such as removing some equations from the public key – destroys the algebraic structure presented by the public key and that was exploited by Gröbner basis algorithms. HFE with removed public equations is often called HFE$^-$, and seems suitable as 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.

All in all, HFE is comparatively in better shape than the SFLASH signature scheme for which polynomial time algorithms are known both to invert [11, 12] and to recover equivalent private keys [18]. SFLASH is based on C^*, hence has a single internal monomial, and the attacks against SFLASH exploit the fact that multiplication matrices commute in some way with this internal monomial. Using this property, 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.

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 have 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). Some instances belonging to this category were proposed by Patarin himself in [30] (an extended version of [29]) 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 may 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 parameter sets. To mount our attack, as in the SFLASH case [11], 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 [22] 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 [20], 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 equivalence for HFE, there is no known attack against the subfield variant of HFE.

As mentioned above, the complexity of nearly all existing attacks on HFE 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 unrealistic if this degree is chosen high enough (a 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 can 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 attack remains polynomial when the degree of the internal polynomial grows polynomially with n.

Organization of the paper. Section 2 gathers some mathematical results, as well as basics on the HFE cryptosystem and known results on the IP Problem. Then, we characterize a class of weak keys in Section 3, and we describe our attack against these weak keys 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 $n > 1$. Recall that $\mathbb{L}$ is essentially the quotient of $\mathbb{K}[X]$ by the principal ideal generated by $P(X)$, an irreducible polynomial of degree n over $\mathbb{K}[X]$. $\mathbb{L}$ is isomorphic to $\mathbb{K}^n$ via an application φ. For the sake of convenience, it can be specified that φ returns the only polynomial of each equivalence class of degree less than n. 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 $\mathbb{L}[X]$. Two matrices A and B

are said to be *similar* if there exist $P \in \mathrm{GL}_n(\mathbb{K})$ such that $A = P \cdot B \cdot P^{-1}$. Lastly, given a fixed element $a \in \mathbb{L}$, the application $x \mapsto a \cdot x$ is linear, and thus can be represented over $\mathbb{K}^n$ by a matrix M_a.

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 an $(n \times n)$ matrix over $\mathbb{K}$ with its polynomial representation over $\mathbb{L}$, and we will refer to the *polynomial representation over* $\mathbb{L}$ of such a matrix. The set of matrices commuting with F over $\mathcal{M}_n(\mathbb{K})$ is the $\mathbb{K}$-vector space of dimension n generated by $(\mathrm{F}^0, \mathrm{F}, \ldots, \mathrm{F}^{n-1})$. We will also need the following lemma:

Lemma 2.1. *Let $M \in \mathrm{GL}_n(\mathbb{K})$ be an invertible matrix, and let $P = \sum_{i=0}^{n-1} a_i \cdot X^i$ (resp. $Q = \sum_{i=0}^{n-1} b_i \cdot X^i$) the polynomial representation of M over $\mathbb{L}$ (resp. M^{-1}). In general the a_i's and b_i's live in $\mathbb{L}$. But we have:*

$$(a_0, a_1, \ldots, a_{n-1}) \in \mathbb{K}^n \iff (b_0, b_1, \ldots, b_{n-1}) \in \mathbb{K}^n.$$

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

HFE basics. The HFE scheme was designed in [29] by Patarin. Note that specific variations of HFE do exist, but we will focus on the basic HFE scheme. Let us briefly recall its mechanism.

Let $\mathbb{K} = \mathbb{F}_q$. 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:

$$\mathbf{f}(X) = \sum_{\substack{0 \leq i,j \leq n \\ q^i + q^j \leq d}} a_{i,j} \cdot X^{q^i + q^j} + \sum_{\substack{0 \leq k \leq n \\ q^k \leq d}} b_k \cdot X^{q^k} + c, \qquad (2.1)$$

with $a_{i,j}$, b_k and c lying in $\mathbb{L}$. Polynomials with the same shape as $\mathbf{f}$ are called HFE polynomials. Note that these polynomials 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. 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} if decryption needs to be practical. However, d must be a growing function of n as simple polynomial-time decryption attacks exist otherwise. We therefore assume that d is a polynomial function of n. It also makes sense to consider degree bounds of the form $d = 2 \cdot q^D$, because in equation (2.1), 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 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 $\mathbf{PK} = T \circ \mathbf{f} \circ S$. In order to offer non-trivial encryption, $\mathbf{f}$ must logically be non-linear. Also, because HFE was designed specifically to circumvent the attack that destroyed C*, we will assume that the internal polynomial always has at least two non-linear terms.

Note that when $\mathbb{K} = \mathbb{F}_2$, we may assume $a_{i,i} = 0$, by choosing b_{i+1} accordingly. As such, there are $D(D+5)/2 + 3$ terms in $\mathbf{f}$ when $q \neq 2$ and $(D+2)(D+1)/2 + 2$ terms when $q = 2$.

Equivalent keys for HFE. In HFE, the public key can be derived from the secret key in polynomial time by an algorithm PKGen that takes as argument $T, \mathbf{f}, S$ and $\mathbb{L}$ (i.e., the irreducible polynomial P defining $\mathbb{L}$ and the correspondence φ between $\mathbb{L}$ and $\mathbb{K}^n$). Two secret keys are *equivalent* if they yield the same public key. For instance, it is shown in [40, 41] that an HFE public-key is always generated by a secret key in which S and T are linear (as opposed to affine). The affine part of S and T can be removed by changing the constant component of $\mathbf{f}$. Next, if $\alpha, \beta \in \mathbb{L}$, then it is possible to simultaneously replace T by $T \cdot M_\alpha$ and S by $M_\beta \cdot S$. It is sufficient to replace $\mathbf{f}$ by $\alpha \cdot \mathbf{f}(\beta^{-1} \cdot X)$ in order for the public key to remain the same, and this allows to choose the values of both S and T on one point.

As a consequence, a set of $q^{2n} \cdot (q^n - 1)^2$ equivalent secret keys is identified (this number assumes that $\mathbb{L}$ is fixed). It was not formally established that all the equivalent secret keys belong to this set, even though this seems likely when $\mathbf{f}$ is a randomly-chosen HFE polynomial.

Irrelevance of keeping the extension field secret. While the original description of HFE [29] explicitly specifies that the extension field $\mathbb{L}$ must be part of the secret key, the same paper notes that this does not improve the security of the trapdoor, because there always exist equivalent secret keys for all the possible descriptions of $\mathbb{L}$. As a matter of fact, the specifications of both Quartz [32] and SFLASH [31] make the extension field public. In any case, it is possible to generate the *same* public key from the *same* secret polynomial, while fixing an arbitrary irreducible polynomial P defining $\mathbb{L}$, and an arbitrary correspondence between $\mathbb{L}$ and $\mathbb{K}^n$. Any isomorphism between $\mathbb{K}^n$ and $\mathbb{L}$ being an invertible $\mathbb{K}$-linear map [24], it simply requires slight modifications on S and T.

Proposition 2.2. *Let* $\mathbf{SK} = (T, \mathbf{f}, S, P, \varphi)$ *be an HFE secret key. Then for any choice of an extension field* $\mathbb{L}' = \mathbb{K}[X]/P'(X)$ *of degree n, and for any choice of an isomorphism* φ' *between* $\mathbb{L}'$ *and* $\mathbb{K}^n$*, there exist two affine bijections* S' *and* T' *such that* $\mathbf{SK}' = (T', \mathbf{f}, S', P', \varphi')$ *is equivalent to* $\mathbf{SK}$ *(i.e., generates the same public key).*

Proof. Recall that all finite fields of the same cardinality are isomorphic [24]. Therefore let us consider a field isomorphism $\zeta : \mathbb{L} \to \mathbb{L}'$. Recall that $\varphi : \mathbb{K}^n \to \mathbb{L}$ and $\varphi' : \mathbb{K}^n \to \mathbb{L}'$ are both isomorphisms as well. The notation $\mathbf{PK} = T \circ \mathbf{f} \circ S$ is unambiguous when the extension field $\mathbb{L}$ is clearly defined. Here we will write

$$\mathbf{PK} = T \circ \varphi^{-1} \circ \mathbf{f}_{\mathbb{L}} \circ \varphi \circ S,$$
$$\mathbf{PK}' = T' \circ \varphi'^{-1} \circ \mathbf{f}_{\mathbb{L}'} \circ \varphi' \circ S'.$$

Let us solve $\mathbf{PK} = \mathbf{PK}'$ for S' and T'. Because the two internal polynomials in $\mathbf{PK}$ and $\mathbf{PK}'$ are the same, we can write

$$\varphi \circ T^{-1} \circ \mathbf{PK} \circ S^{-1} \circ \varphi^{-1} = \zeta^{-1} \circ \varphi' \circ T'^{-1} \circ \mathbf{PK}' \circ S'^{-1} \circ \varphi'^{-1} \circ \zeta.$$

It follows that in order to enforce $\mathbf{PK} = \mathbf{PK}'$ it is sufficient to have

$$T' = T \cdot (\varphi'^{-1} \circ \zeta \circ \varphi)^{-1},$$
$$S' = S \cdot (\varphi'^{-1} \circ \zeta \circ \varphi).$$

Since $\varphi'^{-1} \circ \zeta \circ \varphi$ is an automorphism of $\mathbb{K}^n$, we have $S', T' \in \mathrm{GL}_n(\mathbb{K})$. Thus the secret key $(T', \mathbf{f}, S', P', \varphi')$ is equivalent to $(T, \mathbf{f}, S, P, \varphi)$. □

So, keeping the representation of $\mathbb{L}$ secret does not improve the resistance of HFE to key-recovery attacks. 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.

2.3 Known algorithms for finding isomorphisms of polynomials

In this section we briefly list the known techniques to solve the IP Problem. This problem was first introduced in [29], and its hardness underlies for instance the hardness of the key-recovery of the C^* scheme. As already mentioned, the security of HFE does not rely in general on the hardness of this problem. However, 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. Moreover, solving this problem happens to be practical in some cases (e.g. the "subfield" case, see Section 5).

Recall that finding an "isomorphism" between two vectors of multivariate polynomials **a** and **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

$$\mathbf{b}(x) = V(\mathbf{a}(U \cdot x + c)) + d. \tag{2.2}$$

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 [34], for which no algorithm with polynomial worst-case complexity is known.

The first non-trivial algorithm for IP, known as the "To and Fro" technique, is due to Courtois et al. [34]. In its primitive form, this algorithm assumes the ability to invert 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 $\mathcal{O}(q^{n/2})$ 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 random inhomogeneous instances of IP (where the coefficients of all the monomials of all degree of **a** and **b** are randomly chosen in $\mathbb{F}_q$). On these instances of IP, the practical complexity of [17] has empirically been observed to be $\mathcal{O}(n^9)$.

More recently, a faster algorithm dealing with the same class of instances (i.e., $c = d = 0$) provably achieved an expected complexity of $\mathcal{O}(n^6)$ 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 **a** and **b** are homogeneous, and these are the most frequent settings in multivariate cryptography. However, it is 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 with these hard instances in time $\mathcal{O}(n^{3.5} \cdot q^{n/2})$.

3 A specific family of HFE secret polynomials

Similarly to the attacks against C^* or SFLASH, the main idea we exploit is that some HFE secret polynomials may *commute* with some special functions. This commutativity property can then in turn be used to acquire information on the secret elements.

A commutativity property for some HFE secret polynomials. Let us first consider the à la C^* case, where the secret polynomial $\mathbf{f}$ over $\mathbb{L}$ is just a monomial $\alpha \cdot X^{q^i+q^j}$, with $\alpha \in \mathbb{L}$. Then the public key $\mathbf{PK} = T \circ \mathbf{f} \circ S$ can also be written as $(T \cdot M_\alpha) \circ X^{q^i+q^j} \circ S$, by "absorbing" the multiplication by the constant α into the outer secret linear transformation. As a consequence, without loss of generality, we can assume that $\alpha = 1$.

This secret monomial has very special commutativity properties, which were used in [11,12] to break SFLASH. More precisely, composing it on the right-hand side by multiplications M_x by an element $x \in \mathbb{L}$ is equivalent to composing it on the left-hand side by $M_{x^{q^i+q^j}}$. Another property, not used in [11,12], is that it also commutes with the Frobenius map F (and hence with the iterates of the Frobenius map).

When we consider an arbitrary HFE secret polynomial, the two commutation properties no longer hold in general. However, if we restrict the HFE polynomials to have their coefficients in $\mathbb{K}$ (instead of the extension field $\mathbb{L}$), we lose commutativity with multiplications but commutativity with the Frobenius map still remains. In the sequel, we show how this specific property can be exploited to perform a key-recovery attack, described in Section 4. Therefore, we will say the secret keys in which the internal polynomial has coefficients in $\mathbb{K}$ are *weak secret keys*. Such instances of HFE are illustrated by Figure 1.

Our key-recovery attack could also apply to monomial instances of HFE, but this is not the point of this paper, as it has already been efficiently done [11,12,18].

Corresponding public keys. The attack we discuss in this paper recovers a useful secret key by exploiting only knowledge of the public key. The attack works at the sole condition that the public key can be generated by a weak secret key. Therefore, we will say that such public keys are *weak public keys*. We note that a weak public key has not necessarily been generated by a weak secret key. For instance, let $\mathbf{f}$ be an HFE polynomial with coefficients in $\mathbb{K}$, and let $\alpha, \beta \in \mathbb{L}$.

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

Figure 1. A weak public-key $\mathbf{PK} = T \circ \mathbf{f} \circ S$. The broken arrow indicates that $\mathbf{f}$ has coefficients in $\mathbb{K}$.

Then let us define $\mathbf{f}' : x \mapsto \alpha \cdot \mathbf{f}(b \cdot x)$. This HFE polynomial has coefficients in $\mathbb{L}$. Now let $S, T \in \mathrm{GL}_n(\mathbb{K})$, and consider the HFE secret key $(T, \mathbf{f}', S)$. It does not fall into our definition of weak secret keys, because $\mathbf{f}' \notin \mathbb{K}[X]$, but it generates a weak public key: it is straightforward that it is equivalent to $(T \cdot M_\alpha, \mathbf{f}, M_\beta \cdot S)$. To summarize, our attack will be applicable whenever the internal polynomial has the following shape:

$$\mathbf{f}'(X) = \sum_{\substack{0\le i,j\le n\\ q^i+q^j\le d}} \left(\alpha \cdot u_{i,j} \cdot \beta^{q^i+q^j}\right) \cdot X^{q^i+q^j} + \sum_{\substack{0\le k\le n\\ q^k\le d}} \left(\alpha \cdot v_k \cdot \beta^{q^k}\right) \cdot X^{q^k} + \alpha \cdot c \tag{3.1}$$

with $u_{i,j}, v_k, c \in \mathbb{K}$, $\alpha, \beta \in \mathbb{L}$. In other words $\mathbf{f}' = M_\alpha \circ \mathbf{f} \circ M_\beta$, where $\mathbf{f}$ has coefficients in $\mathbb{K}$.

Notice that legitimate users could easily check whether their secret key generates a weak public key by checking if their internal polynomial can be written as in equation (3.1).

4 The attack

We now describe a key-recovery attack against the class of weak keys described in Section 3. In the sequel, we then assume that the internal secret polynomial $\mathbf{f}$ has coefficients in $\mathbb{K}$.

As the attack is quite complex, let us give an overview. A pseudo-code of the attack is given in Algorithm 1. First, as already mentioned in Section 3, we use the commutation of the Frobenius map with the secret polynomials considered, which propagates to the public key $\mathbf{PK}$. This key property allows us to recover applications closely related to S and T. An interpolation of $\mathbf{PK}$ 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 an equivalent one that allows us

Algorithm 1 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 $\deg \mathbf{f}' \leq \deg \mathbf{f}$.
Let F denote the matrix of the $X \mapsto X^q$ function over $\mathbb{K}^n$.
// Section 4.1
repeat
    Let $U, V \in \mathrm{GL}_n(\mathbb{K})$ be a (random) solution to the IP Problem:
        $$U \circ \mathbf{PK} = \mathbf{PK} \circ V.$$
until there exists $P \in \mathrm{GL}_n(\mathbb{K})$ such that $U = P^{-1} \cdot \mathrm{F} \cdot P$
// Section 4.2
for all $i_0$ in $[1; n-1]$ co-prime with $n$ do
    Let $k = {i_0}^{-1} \mod n$.
    Compute $\widetilde{S}, \widetilde{T}$ such that $\mathrm{F} = \widetilde{S} \cdot V^k \cdot \widetilde{S}^{-1} = \widetilde{T}^{-1} \cdot U^k \cdot \widetilde{T}$.
    // Section 4.3
    Interpolate $\mathbf{g} = \widetilde{T}^{-1} \cdot \mathbf{PK} \cdot \widetilde{S}^{-1}$.
    if $\mathbf{g}$ has all coefficients in $\mathbb{K}$ then
        // Section 4.4
        Compute $F_1, F_2 \in \mathrm{GL}_n(\mathbb{K})$ and $\mathbf{f}_2 \in \mathbb{K}[X]$, s.t. $\deg \mathbf{f}_2 \leq \deg \mathbf{f}$ and:
            $$\mathbf{g} \circ F_1 = F_2^{-1} \circ \mathbf{f}_2.$$
        return $\left(\widetilde{T} \cdot F_2^{-1}, \mathbf{f}_2, F_1^{-1} \cdot \widetilde{S}\right)$
    end if
end for
```

to decrypt as efficiently as the secret key owner. All these assertions are made explicit and justified in this section.

4.1 A useful property of HFE secret polynomials lying in $\mathbb{K}[X]$

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

$$\mathbf{f} \circ \mathrm{F}(X) = \mathrm{F} \circ \mathbf{f}(X). \tag{4.1}$$

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 in the public-key.

Proposition 4.1. *There exist 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:*

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

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

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

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

Then, because of (4.1), $\psi(F), \ldots, \psi(F^{n-1})$ are automorphisms of the public key. □

The existence of other solutions besides those mentioned in Proposition 4.1 is extremely unlikely, unless the situation is very degenerate. 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 $\psi(F^{i_0})$, for some unknown power i_0.

Hardness of the IP Problem. We discussed algorithms for solving the IP Problem in Section 2.3. In order for the polynomial-time IP algorithms to apply, the following conditions needs to be met:

(i) The secret transformations S and T must be linear (as opposed to affine).

(ii) The $\mathbb{K}$-linear coefficients b_k of (3.1) must not all be zero.

(iii) The constant coefficient c of (3.1) must be 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 the size of the private key. Second, as shown in Section 2.2, S and T can be assumed to be linear in the usual setting, because of the existence of equivalent keys. However, we stress that this last fact is *no longer true* if the internal polynomial $\mathbf{f}$ is chosen in $\mathbb{K}[X]$ instead of $\mathbb{L}[X]$. Series 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 S and T are linear, and if $c = 0$ in (2.1), 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 $q^{n/2}$. In the case of the "subfield variant" though, all the numerical quantities lie in a subfield $\mathbb{F}_{q'}$ of $\mathbb{F}_q$, where q' is quite small (the typical value is $q' = 2$). This makes breaking the IP instances feasible in practice for the subfield variant (see Section 5.2).

4.2 Retrieving "nearly S" and "nearly T" applications

Let us assume that we have found an automorphism $(U, V) = \psi(\mathrm{F}^{i_0})$ 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" as much information as possible 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 are relatively prime. This latter condition can be easily checked: F^i and F^j are similar matrices 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 this turns out not to be the case, we take another automorphism of **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 derive 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 4.3.

With the preceding notations, we have the following result:

Proposition 4.2. *Let* $(U, V) = \psi(\mathrm{F}^{i_0})$, *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* $\mathrm{F} = \widetilde{S} \cdot V^k \cdot \widetilde{S}^{-1}$ *and* $\mathrm{F} = \widetilde{T}^{-1} \cdot U^k \cdot \widetilde{T}$.

(ii) *Both* $\widetilde{S} \cdot S^{-1}$ *and* $\widetilde{T} \cdot T^{-1}$ *commute with* F, *hence their polynomial representations over* $\mathbb{L}$ *are in fact polynomials with coefficients in* $\mathbb{K}$.

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 $\mathrm{F}^{i_0 \cdot k} = \mathrm{F}$.

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

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

Thus $\mathrm{F} \cdot \widetilde{S} \cdot S^{-1} = \widetilde{S} \cdot S^{-1} \cdot \mathrm{F}$. This commutation property directly implies the announced result on the polynomial representations (cf. Section 2). □

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.

4.3 Building a high-degree equivalent secret key over $\mathbb{K}[X]$

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. Following Proposition 4.2, we define $F_1 = \widetilde{S} \cdot S^{-1}$ and $F_2 = T^{-1} \cdot \widetilde{T}$, and immediately obtain

$$\begin{aligned} \widetilde{T}^{-1} \circ \mathbf{PK} \circ \widetilde{S}^{-1} &= F_2^{-1} \circ T^{-1} \circ T \circ \mathbf{f} \circ S \circ S^{-1} \circ F_1^{-1} \\ &= F_2^{-1} \circ \mathbf{f} \circ F_1^{-1}. \end{aligned} \tag{4.2}$$

This seems interesting, and we therefore define

$$\mathbf{g} = \widetilde{T}^{-1} \circ \mathbf{PK} \circ \widetilde{S}^{-1} \bmod (X^{q^n} - X). \tag{4.3}$$

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 $\mathcal{O}(n^2)$ coefficients, and they can be uniquely[1] determined in polynomial time by interpolation; this was noted in [23]. By doing so, we obtain an equivalent secret key, namely $(\widetilde{T}, \mathbf{g}, \widetilde{S})$.

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:

[1] Note that there would not be a unique solution if we did not perform the modular reduction of $\mathbf{g}$.

Proposition 4.3. *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 4.2 (ii) 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 2.1). This and (4.2) show that $\widetilde{T}^{-1} \circ \mathbf{PK} \circ \widetilde{S}^{-1}$ has coefficients in $\mathbb{K}$. From there, it is straightforward that $\mathbf{g}$ has coefficients in $\mathbb{K}$. □

The result of Proposition 4.3 is illustrated in Figure 2. This figure also helps remembering how the applications introduced so far intervene.

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

Figure 2. $\mathbf{PK} = T \circ \mathbf{f} \circ S = \widetilde{T} \circ \mathbf{g} \circ \widetilde{S}$. Broken arrows stand for applications with coefficients in $\mathbb{K}$.

Proposition 4.3 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.

4.4 Recovering a low-degree equivalent secret key

To be usable, 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 (4.2) of Section 4.3. 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 overdetermined 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, where all the coefficients in the expressions are now known to lie in $\mathbb{K}$:

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

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

$$\mathbf{g}(X) = \sum_{q^i+q^j<q^n} a_{ij} X^{q^i+q^j} + \sum_{i=0}^{n-1} b_i X^{q^i} + c,$$

$$\mathbf{f}_2(X) = \sum_{q^i+q^j\le d} e_{ij} X^{q^i+q^j} + \sum_{q^i\le d} f_i X^{q^i} + g.$$

Then, we consider the following polynomial equation, also represented by Figure 3, obtained by composing both sides of equation (4.2) of Section 4.3 with F_1:

$$\mathbf{g} \circ F_1 = F_2^{-1} \circ \mathbf{f}_2. \tag{4.4}$$

$$\begin{tikzcd}
\mathbb{K}^n \arrow[r, dotted, "\mathbf{f}_2"] \arrow[d, dotted, "F_1"'] & \mathbb{K}^n \\
\mathbb{K}^n \arrow[r, "\mathbf{g}"] & \mathbb{K}^n \arrow[u, dotted, "F_2"']
\end{tikzcd}$$

Figure 3. $\mathbf{g} = F_2^{-1} \circ \mathbf{f}_2 \circ F_1^{-1}$. Broken arrows stand for applications with unknown coefficients.

Let us now substitute the expression of F_1 and $\mathbf{g}$ in the left-hand side of (4.4). We find:

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

We observe that $\mathbf{g} \circ F_1$ is a polynomial whose coefficients are quadratic polynomials in the coefficients of F_1. Now, let us substitute the expression of F_2^{-1} and

$\mathbf{f}_2$ in the right-hand side of (4.4). We find:

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

We again find that $F_2^{-1} \circ \mathbf{f}_2$ is a polynomial whose coefficients are quadratic polynomials in the coefficients of both $\mathbf{f}_2$ and F_2^{-1}.

This shows that (4.4) is equivalent to a system of multivariate quadratic equations over $\mathbb{K}$ with $\mathcal{O}(n^2)$ quadratic equations and $\mathcal{O}(n + D^2)$ unknowns, the unknowns being the coefficients of F_1, F_2^{-1} and $\mathbf{f}_2$. This system can be generated by reducing both sides of (4.4) modulo $X^{q^n} - X$ and identifying the coefficients of the monomials in X. The problem of computing the decomposition of equation (4.4) is therefore reduced to that of solving an overdetermined system of quadratic equations.

However, equation (4.4) admits many parasitic solutions (for example, $F_1 = \mathbf{f}_2 = 0$, F_2^{-1} being arbitrary). To avoid these trivial solutions, we in fact consider an extended system:

$$\begin{cases} F_1 \circ F_1^{-1} = \mathrm{Id}, \\ F_2 \circ F_2^{-1} = \mathrm{Id}, \\ \quad \mathbf{g} \circ F_1 = F_2^{-1} \circ \mathbf{f}_2. \end{cases} \tag{4.5}$$

This new system avoids the parasitic solutions by forcing F_1 and F_2 to be invertible. We now argue that (4.5) is also equivalent to a system of quadratic equations, whose unknowns are the coefficients of F_1, F_1^{-1}, F_2, F_2^{-1} and $\mathbf{f}_2$. The third equation has already been shown to be translatable to multivariate quadratic equations. Substituting the definitions of F_1 and F_1^{-1} in $F_1 \circ F_1^{-1}$ yields

$$\sum_{k=0}^{n-1} \sum_{\ell=0}^{n-1} (x_k \cdot y_\ell) \cdot X^{q^{\ell+k}} = X.$$

We observe that the coefficients of the left-hand side are quadratic in the x_k's and y_k's, therefore we obtain n quadratic equations by reducing the LHS modulo $X^{q^n} - X$ and equating the coefficients on both sides of the equation. The same goes for $F_2 \circ F_2^{-1} = \mathrm{Id}$.

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 (4.2) of Section 4.3, as long as we picked the right power of the Frobenius matrix in Section 4.1. In fact, even though we just need one, we know that many solutions exist: for instance because the Frobenius commutes with everything in equation (4.4), 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 $\mathcal{O}(n^2)$ equations in $\mathcal{O}(n + D^2)$ 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. *The Gröbner basis of a system of random quadratic equations with the same number of variables 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 $\mathcal{O}(n^{24})$ by the F4 or F5 algorithm* [14, 15]*. This is true if D is fixed, or even if it grows polynomially with* $\log n$.

Justification of the conjecture. We argue that the complexity of computing a Gröbner basis of our equations is in 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 overdetermined 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 [1,7]. 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 $I \subset k[x_1, \dots, x_m]$, then the complexity of computing a Gröbner basis of I using the F5 algorithm [15] is upper bounded by

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

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 systems 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 becomes 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 $f(z) = (1-z^2)^m/(1-z)^n$, up to the degree of regularity. The degree of regularity is therefore 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, for any constant α 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:

$$\begin{aligned} d_{\text{reg}} &= n\Big(\alpha - \frac{1}{2} - \sqrt{\alpha(\alpha-1)}\Big) - \frac{a_1}{2(\alpha(\alpha-1))^{\frac{1}{6}}} n^{\frac{1}{3}} \\ &\quad - \Bigg(2 - \frac{2\alpha-1}{4\sqrt{\alpha(\alpha-1)}}\Bigg) + \mathcal{O}\big(1/n^{\frac{1}{3}}\big), \quad \text{with } a_1 \approx -2.33811. \end{aligned} \tag{4.6}$$

While we are well aware that it is not theoretically justified (because equation (4.6) is established for a constant α), we now set $\alpha = \beta n$, and express d_{reg} as a function of β. This yields

$$d_{\text{reg}} = \frac{1}{8\beta} - \frac{a_1}{2\beta^{1/3}} - \frac{3}{2} + \mathcal{O}(1/n). \tag{4.7}$$

This heuristic result can be empirically checked to be rather precise, for various values of β and n, as shown in Figure 4. When n grows to infinity, it seems that the degree of regularity converges to a constant, an approximation of which is given by (4.7). We now apply this result to our setting:

(i) Consider that D is fixed. Then when n becomes big, we have $\beta = 1/32$. Equation (4.7) then yields $d_{\text{reg}} = 7$ 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 $\mathcal{O}(n^{8\omega})$.

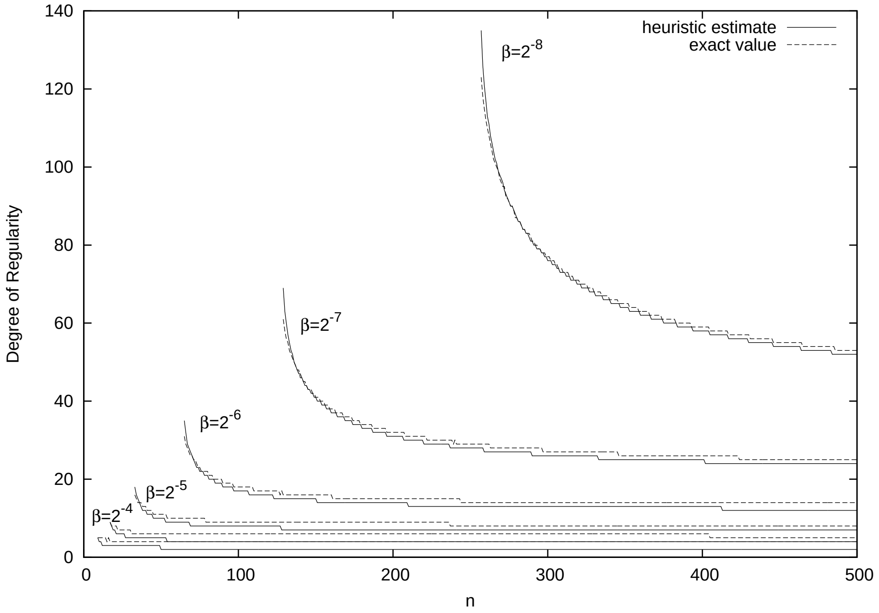

Figure 4. Comparison between the heuristic estimate and the actual values of the degree of regularity for $\beta \cdot n^2$ quadratic equations in n unknowns.

(ii) Consider that the degree of $\mathbf{f}$ grows polynomially with n, which means that $D = \mathcal{O}(\log n)$. In that case we have $\beta = 1/32 + \mathcal{O}(\log n/n)$, and equation (4.7) still yields $d_{\text{reg}} \approx 7$ 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. Figure 5 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 solutions, 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 $\mathcal{O}(n^9)$. Our equations are thus *easier* to solve than random systems with the same parameters.

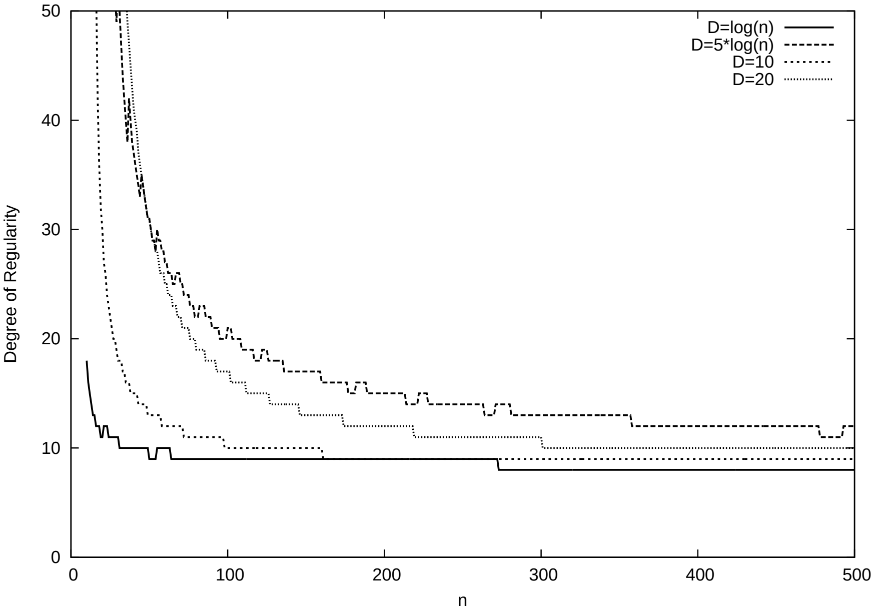

Figure 5. Degree of regularity of semi-generic systems of $n(n+3)/2+1$ quadratic equations in $4n + D(D+5)/2 + 4$ variables.

The solution of the equations yields 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 $\mathbf{f}$.

5 The attack in practice

This section discusses the effectiveness of the attack described in Section 4. First, we estimate the cardinality of the family of secret polynomials concerned by the attack, thus the probability to generate a "weak" public key. We will see that this probability is actually negligible. However, as already discussed in Section 4.1 and illustrated by Patarin's "subfield" variant of HFE [30], we can keep in mind that the choice of the weak parameters we highlighted in the paper can be made on purpose. In Section 5.2, we show practical results of the attack. We will see that the attack is completely practical when the parameters choice make the IP instance solvable in practice. We will also deal with Patarin's "subfield" variant.

5.1 An estimation of the cardinality of the family of secret polynomials

This section shows that the probability that the uniform random choice of a secret key yields a weak public key is negligible. This unfortunate event happens if and only if the internal polynomial $\mathbf{f}$ has the shape given by equation (3.1). We therefore go on to count the number of such polynomials which we call "weak polynomials" for sake of brevity. Let us denote by $\#\,\mathrm{WP}$ the number of weak polynomials, and by $\#\,\mathrm{HFE}(d,n,\mathbb{L})$ the number of HFE polynomials of degree d over $\mathbb{L}$. According to the previous comments (Section 3), a first upper bound is obtained when assuming that for any choice of $\alpha,\beta \in \mathbb{L}^*$ and $\mathbf{f} \in \mathbb{K}[X]$, $M_\alpha \circ \mathbf{f} \circ M_\beta$ is a distinct polynomial. This yields:

$$\#\,\mathrm{WP} \leq \left(q^n - 1\right)^2 \cdot \#\,\mathrm{HFE}(d,n,\mathbb{K}).$$

However we can refine this bound, because some polynomials are counted several times. For instance, if $\pi \in \mathbb{K}$ and $\Pi \in \mathbb{L}$, then $\Pi \cdot \mathbf{f} = (\Pi \cdot \pi^{-1}) \cdot (\pi \cdot \mathbf{f})$. The same goes for right-composition. The bound therefore improves to:

$$\#\,\mathrm{WP} \leq \left(\frac{q^n - 1}{q - 1}\right)^2 \cdot \#\,\mathrm{HFE}(d,n,\mathbb{K}).$$

Note that excluding all elements of $\mathbb{K}$ in our count may not be sufficient as for some specific polynomials $\mathbf{f}$, we still might have counted some polynomials several times. This could eventually happen for some very sparse polynomials $\mathbf{f}$ (made up of two terms for instance), where we could imagine that multiplications by elements belonging to a strict subfield of $\mathbb{L}$ (then defined by the exact expression of the powers of X intervening in $\mathbf{f}$) would commute. However first, this entirely depends on each polynomial, which is why a general better bound is hard to give. And second, this concerns few polynomials and few multiplications, so we can suppose that the previous upper bound is rather good.

Now, in the simpler case where $d = 2q^D$ and $q \neq 2$, we have

$$\#\,\mathrm{HFE}(d,n,\mathbb{K}) = q^{\frac{D(D+5)}{2}+3},$$

and therefore the probability of randomly generating a weak public key is upper bounded by

$$\frac{\#\,\mathrm{WP}}{\#\,\mathrm{HFE}(d,n,\mathbb{L})} \leq \left(\frac{q^n - 1}{q - 1}\right)^2 \cdot \left(q^{\frac{D(D+5)}{2}+3}\right)^{-(n-1)} = \mathcal{O}\left(\left(q^{2-D^2/2}\right)^{n-1}\right).$$

This shows that the probability of generating a weak key out of bad luck is exponentially small in the security parameter. In the same vein, we could obtain a fairly obvious lower bound on this probability by counting only the polynomials of the form $M_\alpha \circ \mathbf{f}$. All in all, the exact number of weak polynomials is delicate to estimate, because it depends on the shape and coefficients of $\mathbf{f}$.

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
$\deg \mathbf{f}$	131 072	131 072	128	131 072	131 072
Coefficients of **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 **f**	10	54	29	10	21
Size of **PK** (bits)	143 616	314 364	461 138	13 485	107 970
IP	polynomial	polynomial	polynomial	$\approx$ 1 s	$\approx$ 4 weeks
Interpolation of **g** (once)	79 s	30 min	140 min	51 s	23 min
Gröbner	7 h	1 day	1 week	45 s	3 h
Variables/ Equations	136/ 593	322/ 4947	423/ 10028	124/ 494	253/ 1889
Memory required	2.1 Gbyte	45 Gbyte	180 Gbyte	350 Mbyte	13.9 Gbyte
Order change	15 s	30 min	4 h	0 s	30 s

Table 1. Timings for the attack.

5.2 Practical applications and experiments

We implemented 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 2.3 Ghz Xeon “Nehalem” computer with 74 Gbyte of RAM. We tested our attack on five sets of parameters described below. We forged the solution of the IP instance from the knowledge of the secret S and T, and estimated the time needed to solve the corresponding problem. The actual timings are given in Table 1.

Weak keys. We first tested the attack on realistically-sized weak keys, corresponding to parameter sets A, B and C. The chosen parameters allows the encryption or signature of 256, 134 and 97 bits, respectively. We chose 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 chose 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 being practical on parameter sets A and B.

Patarin's "subfield" variant of HFE. In order to reduce the size of the public key, Patarin suggests in [30] a "subfield" variant of HFE, in which the coefficients of the quadratic equations of **PK** live in a subfield $\mathbb{k}$ of $\mathbb{K}$. If $\mathbb{K} = \mathbb{F}_{256}$ and $\mathbb{k} = \mathbb{F}_2$, 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 $\mathbb{k}$ (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 $\mathbb{k}$ relatively small. The former implies that D cannot be very huge, otherwise decryption is impractical, while the latter means that there is 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 Table 1. For practical decryption, we have to choose $D = 2$ (yielding an internal polynomial of degree at most 131 072), 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-$\mathbf{f}$-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 [34]. 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 hour.

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.

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 $\mathbb{L}$. This modification includes a suggestion of [30] which remained unbroken until today. 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.

Bibliography

[1] M. Bardet, J.-C. Faugère, B. Salvy and B.-Y. Yang, Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems, in: *MEGA'05*, 8th International Symposium on Effective Methods in Algebraic Geometry, Porto Conte, Alghero, Sardinia (Italy), May 27–June 1, 2005.

[2] W. Bosma, J. J. Cannon and C. Playoust, The Magma algebra system I: The user language, *J. Symb. Comput.* **24** (1997), 235–265.

[3] C. Bouillaguet, J.-C. Faugèe, P.-A. Fouque and L. Pèrret, Isomorphism of polynomials: New results, preprint (2010). www.di.ens.fr/~bouillaguet/pub/ip.pdf.

[4] A. Braeken, C. Wolf and B. Preneel, A study of the security of unbalanced oil and vinegar signature schemes, in: *CT-RSA*, Lecture Notes in Computer Science 3376, Springer (2005), 29–43.

[5] B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, Ph.D. thesis, University of Innsbruck, 1965.

[6] J. F. Buss, G. S. Frandsen and J. Shallit, The computational complexity of some problems of linear algebra, *J. Comput. Syst. Sci.* **58** (1999), 572–596.

[7] D. A. Cox, J. Little and D. O'Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, 3rd edition, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 2007.

[8] D. A. Cox, J. B. Little and D. O'Shea, *Ideals, Varieties and Algorithms*, Springer, 2005.

[9] P. Dembowski and T. G. Ostrom, Planes of order n with collineation groups of order n^2, *Mathematische Zeitschrift* **103** (1968), 239–258.

[10] C. Diem, The XL-algorithm and a conjecture from commutative algebra, in: *ASIACRYPT*, Lecture Notes in Computer Science 3329, Springer (2004), 323–337.

[11] V. Dubois, P.-A. Fouque, A. Shamir and J. Stern, Practical cryptanalysis of SFLASH, in: *CRYPTO*, Lecture Notes in Computer Science 4622, Springer (2007), 1–12.

[12] V. Dubois, P.-A. Fouque and J. Stern, Cryptanalysis of SFLASH with slightly modified parameters, in: *EUROCRYPT*, Lecture Notes in Computer Science 4515, Springer (2007), 264–275.

[13] V. Dubois and N. Gama, The degree of regularity of HFE systems, in: *ASIACRYPT*, Lecture Notes in Computer Science 6477, Springer (2010), 557–576.

[14] J.-C. Faugère, A new efficient algorithm for computing Gröbner bases (F4), *Journal of Pure and Applied Algebra* **139** (1999), 61–88.

[15] J.-C. Faugère, 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, ACM Press, New York (2002), 75–83.

[16] J.-C. Faugère and A. Joux, Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases, in: *CRYPTO*, Lecture Notes in Computer Science 2729, Springer (2003), 44–60.

[17] J.-C. Faugère and L. Perret, Polynomial equivalence problems: Algorithmic and theoretical aspects, in: *EUROCRYPT*, Lecture Notes in Computer Science 4004, Springer (2006), 30–47.

[18] P.-A. Fouque, G. Macario-Rat and J. Stern, Key recovery on hidden monomial multivariate schemes, in: *EUROCRYPT*, Lecture Notes in Computer Science 4965, Springer (2008), 19–30.

[19] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New-York, 1979.

[20] H. Gilbert and M. Minier, Cryptanalysis of SFLASH, in: *EUROCRYPT*, Lecture Notes in Computer Science 2332, Springer (2002), 288–298.

[21] L. Granboulan, A. Joux and J. Stern, Inverting HFE is quasipolynomial, in: *CRYPTO*, Lecture Notes in Computer Science 4117, Springer (2006), 345–356.

[22] A. Kipnis, J. Patarin and L. Goubin, Unbalanced oil and vinegar signature schemes, in: *EUROCRYPT*, Lecture Notes in Computer Science 1592, Springer (1999), 206–222.

[23] A. Kipnis and A. Shamir, Cryptanalysis of the HFE public key cryptosystem by relinearization, in: *CRYPTO*, Lecture Notes in Computer Science 1666, Springer (1999), 19–30.

[24] R. Lidl and H. Niederreiter, *Finite Fields*, Cambridge University Press, New York, 1997.

[25] T. Matsumoto and H. Imai, Public quadratic polynominal-tuples for efficient signature-verification and message-encryption, in: *EUROCRYPT*, Lecture Notes in Computer Science 330, Springer (1988), 419–453.

[26] R. McEliece, *A Public-Key Cryptosystem Based on Algebraic Coding Theory*, DSN Progress Report 42-44, 1978.

[27] O. Ore, Contributions to the theory of finite fields, *Trans. Am. Math. Soc.* **36** (1934), 243–274.

[28] J. Patarin, Cryptoanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88, in: *CRYPTO*, Lecture Notes in Computer Science 963, Springer (1995), 248–261.

[29] J. Patarin, Hidden fields equations (HFE) and isomorphisms of polynomials(IP): Two new families of asymmetric algorithms, in: *EUROCRYPT*, Lecture Notes in Computer Science 1070, Springer (1996), 33–48.

[30] J. Patarin, Hidden fields equations (HFE) and isomorphisms of polynomials(IP): Two new families of asymmetric algorithms, extended version of [29]. `www.minrank.org/hfe.pdf`.

[31] J. Patarin, N. Courtois and L. Goubin, FLASH, a fast multivariate signature algorithm, in: *Topics in Cryptology*, Lecture Notes in Computer Science 2020, Springer (2001), 298–307.

[32] J. Patarin, N. Courtois and L. Goubin, QUARTZ, 128-bit long digital signatures, in: *Topics in Cryptology*, Lecture Notes in Computer Science 2020, Springer (2001), 282–297.

[33] J. Patarin and L. Goubin, Asymmetric cryptography with S-boxes, in: *ICICS*, Lecture Notes in Computer Science 1334, Springer (1997), 369–380.

[34] J. Patarin, L. Goubin and N. Courtois, Improved algorithms for isomorphisms of polynomials, in: *EUROCRYPT*, Lecture Notes in Computer Science 1403, Springer (1998), 184–200.

[35] J. F. Ritt, Prime and composite polynomials, *American M. S. Trans.* **23** (1922), 51–66.

[36] A. V. Sidorenko and E. M. Gabidulin, The weak keys for HFE, in: *7th International Symposium on Communication Theory and Applications* (2003), 239–244.

[37] R. J. Tao and S. H. Chen, Two varieties of finite automaton public key cryptosystem and digital signatures, *Journal of Computer Science and Technology* **1** (1986), 9–18.

[38] J. von zur Gathen, Functional decomposition of polynomials: The tame Case, *J. Symb. Comput.* **9** (1990), 281–299.

[39] J. von zur Gathen, Functional decomposition of polynomials: The wild case, *J. Symb. Comput.* **10** (1990), 437–452.

[40] C. Wolf and B. Preneel, Large superfluous keys in multivariate quadratic asymmetric systems, in: *Public Key Cryptography*, Lecture Notes in Computer Science 3386, Springer (2005), 275–287.

[41] C. Wolf and B. Preneel, Equivalent keys in multivariate quadratic public key systems, *Journal of Mathematical Crypto* **4** (2011), 349–364.

Received June 1, 2010; revised August 1, 2011; accepted August 10, 2011.

Author information

Charles Bouillaguet, Ecole Normale Supérieure, 75005 Paris, France.
E-mail: `charles.bouillaguet@ens.fr`

Pierre-Alain Fouque, Ecole Normale Supérieure, 75005 Paris, France.
E-mail: `pierre-alain.fouque@ens.fr`

Antoine Joux, Direction Générale de l’Armement; and
Université de Versailles, Saint-Quentin-en-Yveline, France.
E-mail: `antoine.joux@m4x.org`

Joana Treger, Agence Nationale de la Sécurité des Systèmes d’Information; and
Université de Versailles, Saint-Quentin-en-Yveline, France.
E-mail: `joana.marim@ssi.gouv.fr`