On the Security of HFE, HFEv- and Quartz*

Nicolas T. Courtois[1], Magnus Daum[2], and Patrick Felke[2]

[1] CP8 Crypto Lab, SchlumbergerSema
36-38 rue de la Princesse, BP 45, 78430 Louveciennes Cedex, France
courtois@minrank.org
[2] Ruhr-Universität Bochum, Postfach 102148, 44780 Bochum, Germany
{Magnus.Daum,Patrick.Felke}@ruhr-uni-bochum.de

Abstract. Quartz is a signature scheme based on an HFEv- trapdoor function published at Eurocrypt 1996. In this paper we study "inversion" attacks for Quartz, i.e. attacks that solve the system of multivariate equations used in Quartz. We do not cover some special attacks that forge signatures without inversion.
We are interested in methods to invert the HFEv- trapdoor function or at least to distinguish it from a random system of the same size. There are 4 types of attacks known on HFE: Shamir-Kipnis [27], Shamir-Kipnis-Courtois [8], Courtois [8], and attacks related to Gröbner bases such as the F5/2 attack by Jean Charles Faugère [15, 16].
No attack has been published so far on HFEv- and it was believed to be more secure than HFE. In this paper we show that even modified HFE systems can be successfully attacked. It seems that the complexity of the attack increases by at least a factor of q^{tot} with tot being the total number of perturbations in HFE. From this and all the other known attacks we will estimate what is the complexity of the best "inversion" attack for Quartz.

Keywords: asymmetric cryptography, finite fields, multivariate cryptanalysis, Gröbner bases, Hidden Field Equation, HFE problem, Quartz, Nessie project.

1 Introduction

The HFE family of trapdoor functions, that have been proposed at Eurocrypt 96 [22] generalizes the previous Matsumoto-Imai cryptosystem called C^* from Eurocrypt 88 [20] broken by Patarin[1] in [21]. The HFE family consists of a so called basic HFE (also called the basic algebraic version of HFE) and modified versions, also called "combinatorial versions of HFE". These modified versions are built on a basic HFE adding the so called "perturbations" that are expected to make attacks harder, but still conserve the existence of the trapdoor.

* The work described in this paper has been partially supported by the French Ministry of Research under RNRT Project "Turbo-signatures".

[1] As was now made public also H. Dobbertin has independently found this attack in '93/94 being employed by BSI-Institute [private communication].

Y.G. Desmedt (Ed.): PKC 2003, LNCS 2567, pp. 337–350, 2003.

For example the Quartz signature scheme that has been submitted to the European Nessie call for cryptographic primitives uses (as a component) such a combinatorial version of HFE, called HFEv-. In this paper we will study the security of HFEv- and Quartz. We will cover all the "inversion" attacks on Quartz, that forge signatures by solving the system of equations that constitutes the public key. We are interested in inverting the HFEv- trapdoor function, not in recovering the secret key of Quartz[2]. In particular we study the "specific" attacks for Quartz (the contrary of "generic" attacks), i.e. attacks that do use the specific algebraic structure of Quartz due to the existence of a trapdoor. This amounts to see whether the public key of Quartz can be distinguished from a random system of **m**ultivariate **q**uadratic equations (MQ). We will study the complexity of the best algorithms known to solve the system of equations of HFE and/or for general MQ systems of the same size. Our goal is also to estimate the security of Quartz given all the known attacks, including properties/attacks that have been conjectured.
There are other "generic" attacks for Quartz (working even if the public key were a random MQ system), that are not based on inversion, see [11]. These are not covered in this paper.

The paper is organized as follows: In the first part we attempt to cover all what is known about the security of HFE and HFEv-, including experimental and conjectured properties. Then we introduce solving polynomial systems with Gröbner bases algorithms, which seems to be the best way to attack HFE or HFEv- so far. We will make numerous simulations with Gröbner bases algorithms to see how the perturbations affect the security of HFE. We will show that perturbated systems can indeed be attacked. Finally we will use our best knowledge to evaluate the security of the trapdoor one-way function used in Quartz against all known "inversion" attacks.

Notations

We use exactly the same notations n, m, h, r, v as in the description of Quartz [25, 26]. h is the size of the extension field on which the internal HFE univariate polynomial is defined and d is the degree of the polynomial. v is the number of added so called "vinegar" variables, r is the number of removed equations and m the number of equations of the public key, i.e. the number of equations after the removal.

2 Known Attacks on HFE and Its Variants

Attacks on "Basic HFE"

At Crypto'99, Shamir and Kipnis presented a structural attack on HFE [27], that reduces the problem of recovering the secret key of HFE to a problem later

[2] The complexity to recover the secret key is obviously at least as much, and therefore we do not need to care about it.

called MinRank, see [8]. At the same time Courtois evaluated the complexity of the Shamir-Kipnis attack and presented two more efficient attacks, see [8].

All these attacks concern only the "basic HFE". They are subexponential. The original Shamir-Kipnis attack is in at least $n^{\mathcal{O}(\log_q^2 d)}$, see [8]. The same attack improved by Courtois, with a better method of solving the involved MinRank problem, gives $n^{3\log_q d+\mathcal{O}(1)}$, see [8]. Both attacks will recover the secret key. The direct attack by Courtois, that only inverts the trapdoor function without recovering the secret key, is much faster in practice. It requires only about $n^{\frac{3}{2}\log_q d+\mathcal{O}(1)}$ computations. Moreover, in [8, 10], a so called "distillation" attack is presented that asymptotically seems to give even about $n^{\frac{3}{4}\log_q d+\mathcal{O}(1)}$. However for values used in practice it did not give better results (except that it uses much less memory).

There are also attacks that apply general methods for solving polynomial systems (by computing Gröbner bases) and seem to be closely related to the direct attack described by Courtois [8, 10]. Recently Faugère has demonstrated a new, and very efficient attack on the basic HFE. With his new improved algorithm F5/2, he was able to break the so called "HFE Challenge 1" in 96 hours on a 833 MHz Alpha workstation with 4 Giga-bytes of memory, see [16, 15], instead of 2^{62} for the direct attack by Courtois from [8].

In [13] and on page 28 of [12] it is shown experimentally that the complexity of the Buchberger algorithm applied to HFE systems depends very strongly on the degree d of the hidden polynomial or rather on the value $\lceil \log_q(d) \rceil$, exactly as expected from the Courtois attacks [8, 10]. This observation was also confirmed by Faugère with F5/2 (private communication).

State of the Art on the Modified Versions of HFE

Until now no attacks have been published on HFE- or other modified versions of HFE, except two attacks on C^{*--} published in [24] and [17]. Though none of the above described attacks on HFE has been specifically designed to work on HFEv−, it is important to see that the direct attack by Courtois and all the Gröbner bases attacks can be applied to any system of equations, without adaptation. For the other attacks, it is not known if they can be extended or adapted. At first sight, from some simulations made by Courtois in [8, 10], it seems that his attack does not work at all, even if we apply one single perturbation "minus" on an HFE-system. From this, if all the known attacks on HFE have indeed a common ground, one might think that the modified "combinatorial" versions of HFE are much more secure than the original HFE, and even Faugère's attack (the fastest known today) will fail, i.e. the complexity will be largely increased. But the questions is, how much will it be increased?

Extrapolating from the complexity of two existing attacks on C^{*--}[24, 17], and having in mind some unpublished attacks on HFE- [10], Courtois conjectured that if *tot* is the total number of perturbations used, then the security of a multivariate scheme such as HFE will be increased by at least a factor of q^{tot}. In this formulation "at least" suggests that it is probably much more secure in many interesting cases.

In this paper we show that this optimistic view is not true. We will see that the Gröbner bases algorithms can solve systems modified with the "minus" perturbation, the "vinegar" perturbation and with the combination of both[3]. The perturbations are not an absolute weapon against these attacks. However we will see that the perturbations increase security, by some amount, and this amount seems to be indeed at least q^{tot}, but not much more. Thus the true question to be answered for Quartz, will be: Is the number of perturbations sufficient or not, to achieve the desired security level?[4]

3 General Methods for Solving Systems of Multivariate Polynomial Equations

The HFE family of cryptosystems is based on the hardness of the problem of finding a solution to a system of **m**ultivariate **q**uadratic equations, the so called MQ problem, see [10]. It turns out that the best attacks known up to date for this MQ problem, such as the XL algorithm from Eurocrypt 2000 (see [7]), the aforementioned experimental attacks on HFE by Courtois, and the more advanced attack by Faugère on HFE, are all clearly related. They all can be seen as a way of manipulating the initial equations multiplied by some monomials, that are linearly combined. The language and the tools to deal with such situations are provided by the theory of Gröbner bases, a part of the computational algebraic geometry.

3.1 Solving Systems with Gröbner Bases Algorithms

In this section we describe very briefly some features of solving systems with Gröbner bases. A comprehensive treatment of this theory can be found in [4].

Given a system of m polynomial equations

$$p_i(x_1, \ldots, x_n) = y_i, \ i = 1, \ldots, m$$

with some $p_i \in GF(q)[x_1, \ldots, x_n], y_i \in GF(q)$, the classical way of solving such a system is to consider the ideal $I := (\tilde{p}_1, \ldots, \tilde{p}_m)$ generated by the polynomials $\tilde{p}_i := p_i - y_i$ and to compute the set of all common zeros of these polynomials over the algebraic closure, the so called variety of this ideal. This is done by computing a special ideal basis[5], a so called Gröbner basis[6]. A typical example for a Gröbner

[3] Precisely, an HFEv- that combines 4 "vinegar" perturbations and 3 "minus" perturbations is used in Quartz

[4] Unfortunately this question is critical in Quartz. Due to the very short signatures in Quartz, the total number of perturbations used is only 7 and $q^r = 2^7$. It is very different in Sflash that has $q^{tot} = (2^7)^{11} = 2^{77}$.

[5] An ideal basis is a set of generators for the $GF(q)[x_1, \ldots, x_n]$-module I.

[6] A Gröbner basis is defined with respect to a term ordering. A term ordering generalizes the monomial ordering of univariate polynomials and thus a polynomial division can be defined. A Gröbner basis has some special properties with respect to this division.

basis is $\gcd(\tilde{p}_1, \ldots, \tilde{p}_m)$, if the generators are univariate polynomials. Another well known example consists of the triangular system of linear polynomials which can be computed by Gaussian elimination provided that the generators are linear polynomials.

In general one usually computes a so called lexicographical[7] Gröbner basis of the ideal, which has a triangular structure, similar to that known from systems of linear equations after applying Gaussian elimination. More precisely for each $i = 1, \ldots, n$ there is at least one polynomial in the Gröbner basis with the property that it includes only monomials with variables $x_1, \ldots, x_i$. Therefore one can compute the common zeros by factoring one univariate polynomial, then substituting this partial solution into the polynomial(s) with two variables, and thus getting further univariate polynomials to factor and so on.

The classical algorithm to compute a Gröbner basis is the Buchberger algorithm, see for example [4]. In general this algorithm has double exponential worst case complexity. However in the setting of multivariate cryptography we are only interested in solutions over the base field $GF(q)$ and not in the algebraic closure. In this case the complexity can be cut down to single exponential worst case complexity just by adapting the inputted system:
Since the set of solutions of the equation $x^q = x$ is equal to $GF(q)$, the variety of the enlarged ideal $(\tilde{p}_1, \ldots, \tilde{p}_m, x_1^q - x_1, \ldots, x_n^q - x_n)$ consists just of those solutions of the original system which are lying in the base field. It can be shown that applying Buchberger algorithm to ideals of this form has single exponential worst case complexity (for this see [13] or [1]).

Due to Faugère there are some comparatively very efficient alternatives to the Buchberger algorithm for computing Gröbner Bases (the algorithms F4, see [14], F5 and recently F5/2, see [15, 16]).

3.2 The Special Case of Signing

The common drawback of the algorithms described above is that they compute the complete variety (i.e. the set of all the solutions) and are unable to profit from the fact that we only need one of the solutions. Indeed, in order to forge a signature for a given message in a HFEv- based signature scheme, it is enough to be able to compute at least one solution to a given system of equations[8].

In the case of HFEv- systems, with r perturbations of type "minus" and v of type "vinegar" we obtain a system with $n = h + v$ variables, and $m = h - r$ equations. Such a system has an expected number of about q^{r+v} solutions, and an algorithm that computes a Gröbner basis does too much work. There is however an obvious way to reduce the number of computed solutions. We will substitute $n - m = r + v$ of the variables with some arbitrary fixed values, and

[7] The lexicographical term ordering is the best suited for computing zeros.

[8] Some signature schemes based on HFEv- compute signatures using several inverses, i.e. they need to compute $F^{-1}(y)$ several times in a row for different y, with F being the public key. This allows shorter signatures, see [22, 10] or [11] for explanation. For example in Quartz there are 4 inverses [25, 26].

get a system with $m = h - r$ unknowns and as many equations, which will have about 1 solution on average. Such systems prove to be substantially easier to solve than the original system with many solutions. In Section 4 below we will apply this idea. It seems that fixing exactly $n - m$ variables is the optimal way to solve a system of equations with Gröbner bases, at least when $n > m$, see the Appendix A of the full version ([29]) for further discussion.

4 Applying Gröbner Bases to HFE

To study the effect of the perturbations "minus" and "vinegar" we did many simulations on HFEv- systems with different sets of parameters. We were especially interested in the question how much randomness is added by these perturbations. This is meant to indicate how much perturbations will be needed to produce an HFEv- system, that is indistinguishable from a random system of the same size (with no trapdoor).

The simulations were done using the `stdfglm` function of Singular for computing Gröbner bases (a fast and general implementation of the Buchberger algorithm, see [18]). They were run on a 1.5 GHz Pentium-4 PC, working under Windows 2000. We did simulations on HFE systems of degree $d \in \{5, 9, 17\}$ and on randomly generated systems, both with $h \in \{15, 19, 21\}$, $0 \leq r \leq 3$ and $0 \leq v \leq 5$. We only measured times for systems that have a solution, and systematically casted away systems that have no solution. Following the idea of fixing variables described in Section 3.2 and discussed in Appendix A of the full version ([29]), our simulations apply the following steps:

- Given an HFEv- system with $n = h + v$ unknowns, $m = h - r$ equations, choose $n - m = v + r$ variables to fix.
- Fix the $n - m = v + r$ variables with a set of values not chosen before and solve the resulting system with $m = h - r$ unknowns and equations using Buchberger algorithm (i.e. use the `stdfglm` function of SINGULAR).
- Repeat the fixing and solving until you find a solution to the initial system (one is enough).

5 Our Methodology

Cryptosystems of the HFE family have two independent security parameters: The extension degree h and the degree of the hidden polynomial d. Quite often they also have additional security parameters, for example v and r for HFEv- and Quartz. This makes the study of their security much more complex than for cryptosystems that basically have one security parameter such as RSA. It also makes the multivariate cryptographic schemes much more flexible than the usual schemes. For example one parameter (in our case h) can usually be small to achieve a cryptosystem that operates on small blocks (and e.g. allows short signatures), and the other parameters can be independently adjusted to achieve the desired security level.

5.1 Critical Parameters for Quartz

From the design of Quartz [25, 26], we see that the parameters h, v, r are more or less constrained by the requirement to have very short signatures. Thus the security of Quartz depends mainly on the degree d of the hidden HFE polynomial. As we have already explained in section 2, the complexity of the Gröbner bases algorithms applied to the basic HFE (and also of all other known attacks and for other versions of HFE) does depend very strongly on d. More precisely it depends on the value $\lceil \log_q(d) \rceil$. This is motivated (but not fully explained) by the Shamir-Kipnis attack from [27], and can be seen very distinctively in the Courtois attack [8]. For Gröbner bases algorithms this has been shown experimentally by Daum and Felke, in [12] (also in [13]) and independently confirmed by Faugère for his latest F5/2 algorithm (private communication).

6 The Simulations on HFEv-

Let T_{tot} be the total time (on average) needed by the above described algorithm to find at least one solution of a given system with n variables and m equations; for HFEv- we have $n = h + v$ and $m = h - r$.
Let N_{guess} be the number of guesses (and thus the number of invocations of the Buchberger algorithm) the algorithm had to make till the resulting system (after fixing variables) was solvable. It turns out that the time to solve any of the $N_{guess} - 1$ systems that have no solution and for the last system that has a solution are about the same. Therefore $T \stackrel{def}{=} \frac{T_{tot}}{N_{guess}}$ describes the average time spent by the Buchberger algorithm trying to solve one of the resulting systems (i.e. the initial system after fixing variables).

The Semantics of T: The value T measures the cryptographic quality of each single system with $n - m$ variables fixed. By construction it is independent of $n - m$ (for fixed m), and gives more precise results than just to measure the time of solving the last system. The time to find one solution to the whole system is usually between T and $1.6 \cdot T$, see Appendix A of the full version ([29]).

We define T_{rnd} as the value of T obtained for a random system of quadratic equations of (respectively) the same size. This is our reference time.

6.1 Randomness ∼ Security

The notions of security and randomness are very closely related in cryptography, and even more so for multivariate cryptosystems: The public key of an HFEv- system is a system of multivariate quadratic equations. Therefore breaking an HFEv- system means to solve an instance of the MQ-Problem. At present the hardest instances of the MQ problem we know are just random systems of quadratic equations. It is known that MQ is NP-complete, see [10], which guarantees worst case hardness, and moreover it seems that the problem is difficult on average and even most of the time. Moreover all the known algorithms for this problem are exponential, see for example [13] or [1] for the results on the

complexity of the Buchberger algorithm. In [7] authors raise some hopes for a subexponential algorithm, that would be however very inefficient in practice.

Thus we expect that the complexity to break the HFEv- schemes is always less than to solve a random MQ system of the same size. Moreover, if we increase the parameter d of HFE systems to nearly q^n, the system does converge to a random MQ system. The same is also true for $v \to \infty$ in HFEv-.

We simply measure the time per guess relatively to a random system of the same size, i.e. we define

$$R = \frac{T}{T_{rnd}}.$$

Now we are ready to study how much randomness/security is added by various "perturbations" in HFE.

6.2 The Impact of the Perturbations "Vinegar" and "Minus"

Table 1 shows the times per guess (in seconds) needed for solving HFE systems with $h = 15$ and $d = 5$ perturbated with r times "minus" and v times "vinegar" operations. We also compute our randomness measure R as defined above.

From these values of R (and also the corresponding values in the tables given in Appendix C of the full version ([29])) we can deduce that using perturbations strongly increases the randomness of HFE systems, especially for low degrees d. Moreover we see that the amount of randomness added depends mainly on the total number $v + r$ of used perturbations, and is rather independent of whether one uses "minus" or "vinegar" or a mixture of both.

However if we consider the absolute times needed to solve (find one solution to) the perturbated systems there is an explicit difference between the effects of "minus" and "vinegar":

These times show that applying "minus" in many cases does not increase the absolute time needed to find one solution and may even decrease this time, whereas "vinegar" does increase the attack time. The same behaviour was observed when applying "minus" and "vinegar" to completely random systems.

Table 1. Absolute and relative times per guess for HFEv- systems with $h = 15, d = 5$

Times per Guess $T = T_{tot}/N_{guess}$**:**

		v					
		0	1	2	3	4	5
r	0	0,50	1,93	5,45	11,52	11,89	11,95
	1	0,75	2,36	4,16	5,20	5,23	5,28
	2	0,77	1,30	1,35	1,35	1,38	1,38
	3	0,48	0,48	0,50	0,50	0,51	0,51

Ratios $R = T/T_{rnd}$**:**

		v					
		0	1	2	3	4	5
r	0	0,05	0,18	0,46	0,97	1,01	1,00
	1	0,16	0,50	0,81	1,00	1,00	1,00
	2	0,60	1,00	1,00	1,00	1,00	1,00
	3	1,00	1,01	1,01	1,00	0,99	0,99

(see Appendix C of the full version ([29]) for similar results for different values of d and h)

There we observed that T_{rnd} is nearly independent of v but depends strongly on r. The explanation of this difference is very easy: "minus" in contrast to "vinegar" changes the size of the systems to solve for each guess, from h to $h - r$.

Eventually there are four main conclusions from these simulations:

1. The operations "minus" and "vinegar" are similar in terms of the amount of relative security or randomness added to pure HFE systems of low degree d.
2. For the absolute security they are different, and considering the total time of an attack, "minus" may even decrease the time needed to solve a system, as it decreases the size m of the system to solve after fixing $n - m$ variables. However "minus" still does increase the security against some other, very efficient attacks, for example Patarin's attack on Matsumoto-Imai (C^*) [21].
3. It is unclear, if this is still significant when $m \gg r + v$ as in Quartz.
4. It seems that for the total number of perturbations that can be used, it is more interesting to use as many as possible "vinegar" perturbations and as little as possible "minus" perturbations. However it is not advocated to use only "vinegar" perturbations, the mixture of both might be more secure against some other attacks.

6.3 Some Conclusions for Quartz

It is not directly clear how to quantify the influence of the perturbations in the case of Quartz due to the much bigger size of Quartz systems. It seems that $\log_2 d = \log_2 129$ is a very small degree with respect to the large value of $h = 103$ (as in Quartz). Thus the results of the simulations support the assumption that applying 4 perturbations of type "vinegar" and 3 perturbations of type "minus" will increase the complexity of the attacks. It would be probably better to use 7 perturbations of type "vinegar". However, it is easy to see that this is not possible with the given signature size and security requirements.

There are many constraints in the design of Quartz, and $h = 103$ cannot be easily changed: It is in fact chosen to be a prime, see [25, 26]. Thus with $h = 103$, $v = 7$ and $r = 0$ we get $n = h + v = 110$ and $m = 103$. Since m increases by 3, the length of the signatures would increase from 128 bits to 131. If we instead used $h = 101$ the signature length would be 129 bits, and for $h = 97$ which gives $m = 97$ the scheme would no longer achieve the required security level of 2^{80}, given the well known generic attack in $q^{\frac{4}{5}m}$, see [25, 26, 10, 11].

6.4 Quantitative Effect of "Vinegar" and "Minus"

In this section we will try to see if our simulations confirm (or not) the conjecture from Section 2 to the effect that the security of multivariate schemes with tot perturbations (for us $tot = r + v$) would be increased by (at least) a factor of q^{tot}. We see however that this behaviour can only continue until T achieves T_{rnd}, after that T will not grow anymore. From all our simulations we have the following conclusions:

1. The value R increases quite uniformly with $r + v$, i.e. it is nearly the same for a constant $r + v$ and the same h.
2. All our simulations show that the value R grows at least twice, each time $tot = r + v$ increases by 1, except when R is already very close to 1 and cannot grow anymore.
3. Thus our conjecture that the relative security of multivariate schemes with tot perturbations would be increased by at least a factor of q^{tot} can be said to be confirmed for $R << 1$.
4. Moreover, the smaller R is, the higher is the increase in R when $tot = r + v$ increases by 1. More precisely we see that:
   - For $h = 15$ and $d = 5$ R increases first 4 times, 3 times, 2 times, and then it achieves 1.
   - For $h = 15$ and $d = 9$ R increases first 3 times, 2 times, and then it achieves 1.
   - For $h = 15$ and $d = 17$ R increases first 2 times, and then it achieves 1.
   - For $h = 19$ and $d = 5$ R increases first 8 times, 2 times, 2 times, and then it achieves 1.
   - For $h = 19$ and $d = 9$ R increases first 6 times, 3 times, 2 times, and then it achieves 1.
   - For $h = 19$ and $d = 17$ R increases first 2 times, 2 times, and then it achieves 1.
   - For $h = 21$ and $d = 5$ R increases first 10 times, 12 times, 3 times, 2 times and then it achieves 1.
   - For $h = 21$ and $d = 17$ R increases first 3 times, 2 times, and then it achieves 1.
5. Very clearly, the more $R << 1$, the more is the increase in the relative security R when adding one perturbation. Thus we see that the increase in the relative security R is more significative when d is small, and for a fixed d, it is more significative for a bigger h.

7 Our Estimation of the Security of Quartz

In this section we will first look at the complexity of the best known attacks on HFE when applied to the particular instance of HFE that is a sub-component of the HFEv- that is used in Quartz, i.e. if we ignore the existence of the perturbations. Then we will try to extrapolate what (at best) is the complexity to solve the whole HFEv-.

Attacks on the Internal Sub-Component HFE

Let ω be the exponent of the Gaussian reduction. Though the best known algorithm for this problem is asymptotically in $T^{2.376}$, see [3], the best practical algorithm we know is Strassen's algorithm, see [28]. We will put $\omega = \log_2(7) = 2.807$.

Nobody has ever demonstrated a practical attack on HFE when $d = 129$ as in Quartz. For the so called HFE Challenge 1 described in the extended version

of [22], we have $d = 96$ and two working attacks have been found by Courtois [8] and later Faugère [16]. Their behaviour is similar and we expect that Courtois' attacks give much worse results than Faugère's attack. The complexity of this latter attack has been estimated for $d = 96$ to be $\mathcal{O}(h^8)$ by Faugère [16]. In Appendix B of the full version ([29]) we estimate the constant to be about 1/4. From this we extrapolated that for the basic HFE that can be seen inside the HFEv- used in Quartz, with $d = 129$ instead of $d = 96$ above, the complexity of Faugère's attack should be about $h^{10}/4$, see Appendix B of the full version ([29]) for more details.

Including the Conjectured Effects of the Perturbations

First of all, because of the effect of minus, see point 2 in Section 6.2, we have to use $(h - r)^{10}/4$ instead of $h^{10}/4$. Secondly, since in Quartz there are $tot = 7$ perturbations, following Section 2 and our simulation results in Section 6.4, this complexity should be multiplied by at least 2^7. Thus we obtain that the working factor (WF) needed to inverse a trapdoor one-way function HFEv- should be about:

$$WF(HFEv-) \approx 2^7 \cdot (h-3)^{10}/4 = 2^7 \cdot 100^{10}/4 \approx 2^{71}$$

7.1 Our Estimation of the Security of Quartz

The above figure needs still to be multiplied by 4, because we need to do 4 iterated inversions of the trapdoor function in order to forge a Quartz signature. Moreover in the process of solving we fix $v + r$ variables and we need to repeat the solving process several times, until the system has a solution, 1.6 times on average[9], see Appendix A of the full version ([29]). Thus we have an additional factor of $4 * 1.6$ and we get:

$$WF(Quartz) \approx 2^{74}$$

We convert this to triple-DES operations as required in Nessie project. In order to have a fair comparison, one should implement the triple-DES on the same platform on which Faugère's attack has been implemented. The best optimized 64-bit implementation we are aware of, with improved Biham's bitslice technique, will give about $3 * 192 \approx 2^9$ CPU clocks for triple-DES, see [5]. Thus we get the following estimate:

$$WF(Quartz) \approx 2^{65} \quad \text{TDES computations.}$$

Remark: From Section 6.4, point 4, we see that the factor 2^{v+r} might be in fact bigger, then our complexity would increase.

[9] Indeed, Faugère's attack was applied to a system that had a solution.

8 Is It Necessary to Repair Quartz?

If in Quartz we had $d = 257$, we expect that the attack of Faugère should be in $\mathcal{O}(h^{12})$. Then our complexity will be multiplied by 100^2, i.e. we would get $WF(Quartz) \approx 2^{87}$ which is approximately $\approx 2^{78}$ TDES computations.

8.1 The Speed of Quartz

The best implementation of Quartz we are aware of has been programmed in C by Mehdi-Laurent Akkar for the Nessie project. For the usual Quartz with $d = 129$ (i.e. the revised standard version from [25, 26]), it takes less than 2 seconds to sign a message on a PC working at 2GHz. For $d = 257$ it takes about 6 seconds on average.

9 Conclusion

Quartz is a multivariate signature scheme based on an HFEv- trapdoor function. The interesting property of such functions is that they have two security parameters, and thus one of them can be adjusted independently of the other. Thus it is possible to build signature schemes in which the security is adjusted independently of the signature length. This suggests that in theory, it may be always possible to build a secure signature scheme with fixed signature length, but will it be practical ?

There are several subexponential attacks known for HFE, however no attacks on modified HFE cryptosystems (such as HFEv- used in Quartz) have been published so far. In this paper we showed that it is possible to successfully attack the modified systems by Gröbner bases techniques. From this we tried to evaluate the security of the Quartz signature scheme submitted to Nessie project. The results suggest that the parameter d probably needs to be increased which will accentuate the major drawback of Quartz: it's slowness.

This is currently the price to pay for such a short signature scheme. There are only two other short signature schemes known that give less than the 160 bits of the Weil-pairing scheme [2]. The first one is the McEliece scheme from Asiacrypt 2001, which is about as slow as Quartz and has a much bigger public key of about 1Mbyte instead of 71 Kbytes, see [6]. The second one is the degree 3 Dragon scheme (based on HFE) which seems quite fast, but also has a very big public key, see [23, 19]. It is possible that applying the same Gröbner bases computations to this scheme, the parameter d would have to be revised, and it would end up being quite slow, too.

References

[1] Boo Barkee, Deh Cac Can, Julia Ecks, Theo Moriarty, R. F. Ree: *Why You Cannot Even Hope to use Gröbner Bases in Public Key Cryptography: An Open Letter to a Scientist Who Failed and a Challenge to Those Who Have Not Yet Failed,* in Journal of Symbolic Computation 18, 1994, pp. 497-501 341, 343

[2] Dan Boneh, H. Shacham, and B. Lynn: *Short signatures from the Weil pairing,* Asiacrypt 2001, LNCS 2139, Springer, pp. 514-532. 348
[3] Don Coppersmith, Shmuel Winograd: *Matrix multiplication via arithmetic progressions;* J. Symbolic Computation (1990), 9, pp. 251-280. 346
[4] David Cox, John Little, Donal O'Shea: *Ideals, Varieties, and Algorithms,* Springer-Verlag, 1992 340, 341
[5] Francisco Corella: *A fast implementation of DES and triple DES on PA-RISC 2.0.* `http://www.usenix.org/events/osdi2000/wiess2000/full_papers/corella/corella.pdf` 347
[6] *Nicolas Courtois, Matthieu Finiasz and Nicolas Sendrier: How to achieve a McEliece-based Digital Signature Scheme*; Asiacrypt 2001, LNCS2248, Springer, pp. 157-174. Available at `http://www.cryptosystem.net/mceliece/`. 348
[7] Nicolas Courtois, Adi Shamir, Jacques Patarin, Alexander Klimov, *Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations*, in Advances in Cryptology, Eurocrypt'2000, LNCS 1807, Springer-Verlag, pp. 392-407. 340, 344
[8] Nicolas Courtois: *The security of Hidden Field Equations (HFE)*; Cryptographers' Track RSA Conference 2001, San Francisco 8-12 Avril 2001, LNCS2020, Springer-Verlag, pp. 266-281. 337, 339, 343, 347
[9] Nicolas Courtois: The HFE cryptosystem home page. `http://www.hfe.info`
[10] Nicolas Courtois: *La sécurité des primitives cryptographiques basées sur les problèmes algébriques multivariables MQ, IP, MinRank, et HFE*, PhD thesis, Paris 6 University, 2001, in French. Available at `http://www.minrank.org/phd.pdf`. 339, 340, 341, 343, 345
[11] Nicolas Courtois: *Generic Attacks and the Security of Quartz,* PKC 2003, in these proceedings. A preliminary version was presented at the second Nessie workshop, Royal Holloway, University of London, September 2001. 338, 341, 345
[12] Magnus Daum, Patrick Felke: *Some new aspects concerning the Analysis of HFE type Cryptosystems;* Presented at Yet Another Conference on Cryptography (YACC'02), June 3-7, 2002, Porquerolles Island, France. 339, 343
[13] Magnus Daum: *Das Kryptosystem HFE und quadratische Gleichungssysteme über endlichen Körpern,* Diplomarbeit, Universität Dortmund, 2001. Available from `daum@itsc.ruhr-uni-bochum.de` 339, 341, 343
[14] Jean-Charles Faugère: *A new efficient algorithm for computing Gröbner bases (F_4),* Journal of Pure and Applied Algebra 139, 1–3 (1999) pp. 61-88. See `www.elsevier.com/locate/jpaa` 341
[15] Jean-Charles Faugère: *Computing Gröbner basis without reduction to 0,* technical report LIP6, in preparation, source: private communication. Also presented at the Workshop on Applications of Commutative Algebra, Catania, Italy, 3-6 April 2002. 337, 339, 341
[16] Jean-Charles Faugère: Report on a successful attack of HFE Challege 1 with Gröbner bases algorithm F5/2, announcement that appeared in `sci.crypt` newsgroup on the internet on April 19th 2002. 337, 339, 341, 347
[17] Henri Gilbert, Marine Minier: *Cryptanalysis of SFLASH,* Eurocrypt 2002, LNCS 2332, pp. 288-298, Springer. 339
[18] G.-M. Greuel, G. Pfister, and H. Schönemann. SINGULAR 2.0.3. A Computer Algebra System for Polynomial Computations. Centre for Computer Algebra, University of Kaiserslautern (2001), `www.singular.uni-kl.de`. 342
[19] Neal Koblitz: "Algebraic Aspects of Cryptography"; Springer-Verlag, ACM3, 1998, Chapter 4: "Hidden Monomial Cryptosystems", pp. 80-102. 348

[20] Tsutomu Matsumoto, Hideki Imai: "Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption", Eurocrypt'88, Springer-Verlag 1998, pp. 419-453. 337

[21] Jacques Patarin: "Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88"; Crypto'95, Springer-Verlag, pp. 248-261. 337, 345

[22] Jacques Patarin: "Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms"; Eurocrypt'96, Springer Verlag, pp. 33-48. The extended version can be found at `http://www.minrank.org/hfe.ps` 337, 341, 347

[23] Jacques Patarin: *La Cryptographie Multivariable*; Mémoire d'habilitation à diriger des recherches de l'Université Paris 7, 1999. 348

[24] Jacques Patarin, Nicolas Courtois , Louis Goubin: "C*-+ and HM - Variations around two schemes of T. Matsumoto and H. Imai"; Asiacrypt 1998, Springer-Verlag, pp. 35-49. 339

[25] Jacques Patarin, Louis Goubin, Nicolas Courtois: *Quartz,* **1**28-*bit long digital signatures*; Cryptographers' Track Rsa Conference 2001, San Francisco 8-12 April 2001, LNCS2020, Springer-Verlag.
Note: The Quartz signature scheme has been updated since, see [26]. 338, 341, 343, 345, 348

[26] Jacques Patarin, Louis Goubin, Nicolas Courtois: *Quartz,* **1**28-*bit long digital signatures*; An updated version of Quartz specification available at `http://www.cryptosystem.net/quartz/` 338, 341, 343, 345, 348, 350

[27] Adi Shamir, Aviad Kipnis: "Cryptanalysis of the HFE Public Key Cryptosystem"; Crypto'99. Can be found at http://www.minrank.org/hfesubreg.ps 337, 338, 343

[28] Volker Strassen: *Gaussian Elimination is Not Optimal;* Numerische Mathematik, vol 13, pp 354-356, 1969. 346

[29] Nicolas Courtois, Magnus Daum and Patrick Felke: *On the Security of HFE, HFEv- and Quartz*; Cryptology ePrint Archive, Report 2002/138. Available at `http://eprint.iacr.org`. 342, 343, 344, 347