ASYMMETRIC CRYPTOGRAPHY: HIDDEN FIELD EQUATIONS

Christopher Wolf⋆ and Bart Preneel†

Katholieke Universiteit Leuven
ESAT-COSIC, Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
⋆,†e-mail: {Christopher.Wolf,Bart.Preneel}ATesat.kuleuven.ac.be
web page: http://www.esat.kuleuven.ac.be/cosic/
⋆e-mail: chrisATchristopher-wolf.de

Cryptology ePrint Archive, Report 2004/072

http://eprint.iacr.org/

Current Version: 2005-08-06

First Revision: 2004-06-28
First Version: 2004-03-04

This is the extended version of the article with the same title [WP04].
The original paper has be published at ECCOMAS — European Congress on Computational Methods in Applied Sciences and Engineering;
P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer (eds.); Jyväskylä, 24–28 July 2004 2004.

Key words: Public Key Cryptography, Short Signatures, Multivariate Quadratic Equations, Hidden Field Equations, HFE, Quartz

Abstract. *The most popular public key cryptosystems rely on assumptions from algebraic number theory,* e.g., *the difficulty of factorisation or the discrete logarithm. The set of problems on which secure public key systems can be based is therefore very small:* e.g., *a breakthrough in factorisation would make RSA insecure and hence affect our digital economy quite dramatically. This would be the case if quantum-computer with a large number of qbits were available. Therefore, a wider range of candidate hard problems is needed.*

In 1996, Patarin proposed the "Hidden Field Equations" (HFE) as a base for public key cryptosystems. In a nutshell, they use polynomials over finite fields of different size to disguise the relationship between the private key and the public key. In these systems, the public key consists of multivariate polynomials over finite fields with up to 256 elements for practical implementations. Over finite fields, solving these equations has been shown to be an $\mathcal{NP}$-complete problem. In addition, empirical results show that this problem is *also hard on average, i.e., it can be used for a secure public key signature or encryption scheme.*

In this article, we outline HFE, and its the variations HFE-, HFEv. Moreover, we describe the signature scheme Quartz, which is based on Hidden Field Equations. In addition, we describe the most recent attacks against HFE and sketch two versions of Quartz which are immune against these attacks.

1 INTRODUCTION

1.1 General picture

Public key cryptography is used in e-commerce systems for authentication (electronic signatures) and secure communication (encryption). In contrast to secret key cryptography, public key cryptography has advantages in terms of key distribution. Moreover, signature schemes can not be obtained by secret key schemes. The security of using current public key cryptography for encryption centres on the difficulty of solving certain classes of problem. The RSA scheme relies on the difficulty of factoring very large numbers, while the difficulty of solving discrete logarithms provide the basis for the ElGamal and Elliptic Curve schemes [MvOV96]. Given that the security of these public key schemes rely on such a small number of problems that are *currently* considered hard, research on new schemes that are based on other classes of problems is worthwhile. Such work provides a greater diversity that forces cryptanalysts to expend additional effort by concentrating on a completely new type of problem.

In addition, important results on the potential weaknesses of existing public key schemes are emerging. Techniques for factorisation and solving discrete logarithm continually improve. Polynomial time quantum algorithms can be used to solve both problems [Sho97]; fortunately, quantum computers with more than 7 bits are not yet available and it seems unlikely that quantum computers with 100 bits will be available within the next 10–15 years Nevertheless, this stresses the importance of research into new algorithms for asymmetric encryption and signature schemes.

1.2 Alternative Schemes

In 1996, Patarin proposed the use of a special class of polynomials over finite fields for public key cryptography called "Hidden Field Equations" (HFE) [Pat96]. The scheme supports both encryption and digital signatures; its security is related to the difficulty of solving a random system of multivariate quadratic equations over finite fields. This problem is known to be $\mathcal{NP}$-complete (cf [GJ79, p. 251] and [PG97, App.] for a detailed proof). The HFE scheme generalises and improves the Matsumoto-Imai-system [MI88] which was broken by Patarin [Pat95].

In particular, Hidden Field Equations have been used to construct digital signature schemes, *e.g.*, Quartz [CGP01] and Sflash [CGP03]. Sflash has been developed to suit the smart-card environment; it is a modified and secured version of the Matsumoto-

Imai-system. In this paper, we concentrate on its sister-scheme Quartz, which is based on Hidden Field Equations. In contrast to other schemes, Quartz allows very short signatures, namely only 128 bit (RSA: 1024–2048 bit). The main drawback of Quartz is the size of its public key: 71 kB, which is rather high (RSA: 1024–2048 **bit**). But due to its very short signature size, it is still of high interest for applications with only limited bandwidth for signature transfer. By the construction of Quartz, its signatures are secure against birthday attacks. Moreover, checking the validity of a given signature is very fast. In particular, no crypto-coprocessor is required, even for 8-bit CPUs.

Until recently, signature schemes based on HFE were believed to be secure. The attack of Faugère and Joux — using Gröbner bases (cf [FJ03] and Sect. 4) — raised serious doubts. On the other hand, HFE offer many variations which make it possible to counter attacks (cf Sect. 2.2). In particular, at present it seems possible to vary Quartz in a way that the attack of Faugère and Joux no longer applies. However, more research is needed in this area to obtain secure schemes based on Hidden Field Equations.

1.3 Outline

This paper is organised as follows: first, we give an introduction to Hidden Field Equations (Sect. 2). In Sect. 3 we outline the signature scheme Quartz. An overview of recent attacks against HFE and possible countermeasures is presented in Sect. 4. This paper concludes with Sect. 5.

2 HIDDEN FIELD EQUATIONS

HFE is based on polynomials over finite fields and extension fields. The general idea is to use a polynomial over an extension field as a private key and a vector of polynomials over the underlying finite field as public key. HFE also uses private affine transformations to hide the extension field and the private polynomial. This way HFE provides a trapdoor for an $\mathcal{MQ}$-problem (system of $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic equations).

2.1 Mathematical Background

Fig. 1 gives an outline of the structure of HFE. S and T represent two affine transformations and P is the private polynomial. Hence, the private key is represented by the triple (S, P, T).

The polynomials $(p_1, \ldots, p_n)$ are the public key. These public polynomials as well as the private affine transformations S and T are over $\mathbb{F}$, a finite field with cardinality $q := |\mathbb{F}|$. The private polynomial P is defined over $\mathbb{E}$, an extension field of $\mathbb{F}$ generated by the irreducible polynomial $i(x)$ of degree n.

2.1.1 Encryption and Decryption of Messages using the Private Key

The private polynomial P (with degree d) over $\mathbb{E}$ is an element of $\mathbb{E}[x]$. To keep the public polynomials small, the private polynomial P must have the property that its terms

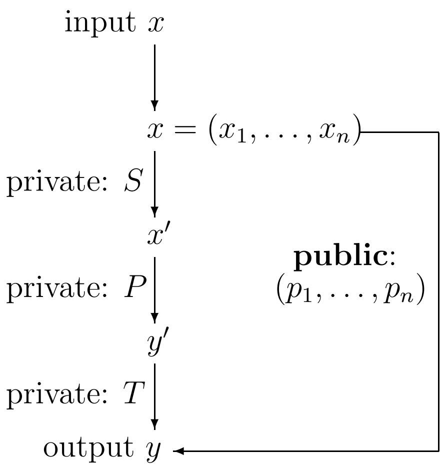

Fig. 1: $\mathcal{MQ}$-trapdoor (S, P, T) in HFE

are at most quadratic over $\mathbb{F}$. In the case of $\mathbb{E} = \mathrm{GF}(2^n)$ this means that the powers have Hamming weight at most 2. In symbols:

$$\begin{array}{rcl} P & : & \mathbb{E} \to \mathbb{E} \\ P(x) & = & \sum c_i x^{h_i}, \text{where} \\ & & c_i \in \mathbb{E}, h_i \le d, h_i \neq h_j \ \forall i \neq j, \\ & & h_i = \begin{cases} 0, & \text{(constant term)} \\ q^a, & a \in \mathbb{N}_0 \\ & \text{(linear terms)} \\ q^b + q^c, & b, c \in \mathbb{N}_0 \\ & \text{(quadratic terms)} \end{cases} \end{array}$$

Since the affine transformations S and T are over $\mathbb{F}$ it is necessary to transfer the message M from $\mathbb{E}$ to $\mathbb{F}^n$ in order to encrypt it (cf Fig. 2). This is done by regarding M as a vector $(x_1, \ldots, x_n) \in \mathbb{F}^n$. Thus we no longer think about the extension field as a field but as an n-dimensional vector-space over $\mathbb{F}$ with the rows of the identity matrix I as basis of $\mathbb{F}^n$. To encrypt $(x_1, \ldots, x_n)$ we first apply S, resulting in x'. At this point x' is transferred from $\mathbb{F}^n$ to $\mathbb{E}$ so we can apply the private polynomial P which is over $\mathbb{E}$. The result is denoted as $y' \in \mathbb{E}$. Once again, y' is transferred to the vector $(y'_1, \ldots, y'_n)$, the transformation T is applied and the final output $y \in \mathbb{E}$ is produced from $(y_1, \ldots, y_n) \in \mathbb{F}^n$.

To decrypt y, the above steps are done in reverse order. This is possible if the private key (S, P, T) is known. The crucial step in the deciphering is not the inversion of S and T, but rather the computation of the solutions of $P(x') = y'$. As P has degree d there are up to d different solutions $X' := \{x'_1, \ldots, x'_d\} \in \mathbb{E}$ for this equation. Addition of redundancy to the message M provides an error-correcting effect that makes it possible to select the

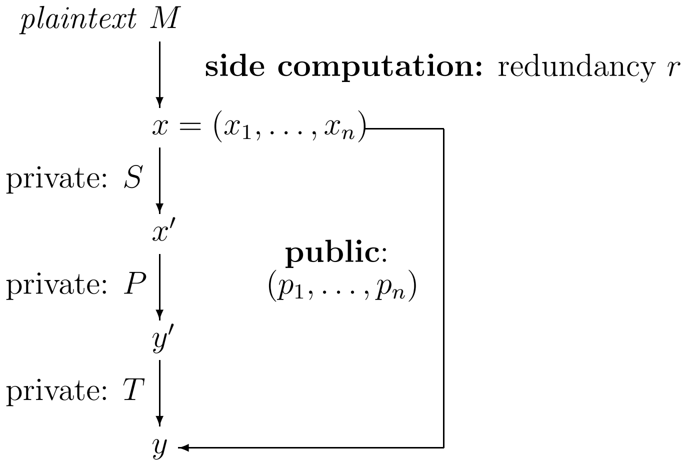

Fig. 2: HFE for encryption of the message M with ciphertext (y, r)

right M from the set of solutions X'. This redundancy is added at the first step (see Fig. 2). Daum provides an estimation of the length of this redundancy [Dau01].

Another way of circumventing this problem would be to take the polynomial P bijective. Unfortunately, Patarin showed that all possible bijections (*e.g.*, Dickson polynomials, Dobbertin polynomials, power functions) lead to very insecure schemes. We refer to [Pat96, Sect. 7.3] for a detailed discussion of this point.

2.1.2 Message Signature

In addition to encryption/decryption, HFE can also be used for signing a message M. As for decryption, we assume that without the trapdoor (S, P, T) it is computationally not feasible to obtain a solution $(x_1, \ldots, x_n)$ for the system of equations

$$\left\{ \begin{array}{rcl} y_1 & = & p_1(x_1, \ldots, x_n) \\ y_2 & = & p_2(x_1, \ldots, x_n) \\ & \vdots & \\ y_n & = & p_n(x_1, \ldots, x_n)\,, \end{array} \right.$$

where $(p_1, \ldots, p_n)$ are quadratic polynomials in the variables $x_1, \ldots, x_n$. In Fig. 3, we follow this notation, so the input for signature generation is denoted with y, while the output is called x. In addition, the message M consists of t elements from $\mathbb{F}$, *i.e.*, $M = (M_1, \ldots M_t) \in \mathbb{F}^t$. The vector $r = (r_1, \ldots, r_f) \in_R \mathbb{F}^f$ is randomly chosen (see below).

If one knows the private key $k = (S, P, T)$, the problem of finding a solution x for given y, reduces to find a solution to the equation $P(x') = y'$ where the polynomial $P \in \mathbb{E}[x]$ has degree d. This is feasible. Unfortunately for HFE, $P(x')$ is usually not a surjection and therefore $\exists y' : P(x') \neq y' \; \forall x' \in \mathbb{E}$. Keeping this in mind, we cannot find a solution

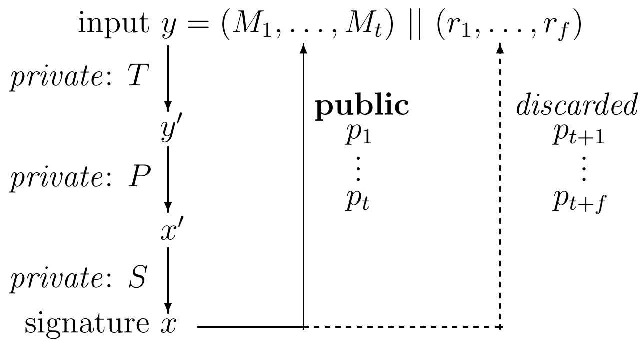

Fig. 3: Signature with MQ, using the HFE trapdoor

$(x_1, \ldots, x_n)$ for each $\mathcal{MQ}$-problem with a HFE trapdoor. So from a practical point of view, if we do not succeed in finding a solution x' for a certain y' in $P(x') = y'$, we have to try another y' until we obtain a result x'. In HFE, the number of y'-values we have to try is small [Pat96]. For a special system such as Quartz (see Sect. 3), we expect to find a solution for one given y' with probability $1 - \frac{1}{e}$, *i.e.*, approx. 60%. However, as Quartz tries up to 128 different values for y' for a given message, the overall probability for not finding any solution drops to approx. 2^{-185} and is therefore negligible [CGP01, p. 9].

For signature generation, we assume that the message $M \in \mathbb{F}^t$ and $n = t + f$. Here, $f \in \mathbb{N}$ is the number of free input variables for the $\mathcal{MQ}$-problem. So $y = (M_1, \ldots, M_t)||(r_1, \ldots, r_f)$ where $\cdot||\cdot$ denotes the concatenation function and $(r_1, \ldots, r_f) \in_R \mathbb{F}^f$ is chosen uniformly at random. The parameter f has to be selected according to the field size of $\mathbb{F}$. As the parameters in the Quartz scheme are $\mathbb{F} = \mathrm{GF}(2)$, and $f = 7$, there are $2^7 = 128$ different y-values for each given message M. In general, we have q^f different y'-values for a given message M. If we can solve the corresponding $P(x') = y'$ for one of these q^f different y-values, we publish the corresponding $x = S^{-1}(x')$ as the signature of M. See Fig. 3 for the overall structure of a signature scheme.

Anybody who wants to verify that the message $m = (m_1, \ldots, m_t)$ was signed by the owner of the private key $K = (S, P, T)$ with $x = (x_1, \ldots, x_n)$, uses the public key, that is, $k = (p_1, \ldots, p_t)$ and compares (denoted $\stackrel{?}{=}$):

$$\begin{aligned} M_1 &\stackrel{?}{=} p_1(x_1, \ldots, x_n) \\ M_2 &\stackrel{?}{=} p_2(x_1, \ldots, x_n) \\ &\vdots \\ M_t &\stackrel{?}{=} p_t(x_1, \ldots, x_n)\,. \end{aligned}$$

If all t equations are satisfied, the signature is valid. Otherwise, it is not. Note that only t of the $m = t + f$ public equations are necessary to verify a signature, the equations $p_{t+1}, \ldots, p_{t+f}$ are not used. Therefore in a signature scheme, only t equations will be published and f equations can be discarded. First, this will lead to a shorter public key. Second, as we will see in Sect. 2.2, this is also expected to improve the security of HFE.

2.1.3 Public Key: Generation and Encryption

After explaining the overall structure of HFE in the previous section, we move on to public key generation. We describe the technique from Matsumoto-Imai [MI88] called "polynomial-interpolation". In [Wol03], Wolf describes a faster way of generating the public key called "base-transformation". Due to space limitations in this paper, we only describe "polynomial-interpolation".

We begin with a description of polynomial interpolation for fields $\mathbb{F} \neq \mathrm{GF}(2)$. The key generation for $\mathbb{F} = \mathrm{GF}(2)$ is slightly different, we deal with this case later in this section. For HFE, we want to obtain polynomials over $\mathbb{F}$ as the public key which have the form

$$p_i(x_1, \ldots, x_n) = \sum_{1 \leq j \leq k \leq n} \gamma_{i,j,k} x_j x_k + \sum_{1 \leq j \leq n} \beta_{i,j} x_j + \alpha_i \,,$$

for $1 \leq i \leq m$ and $\alpha_i, \beta_{i,j}, \gamma_{i,j,k} \in \mathbb{F}$ (constant, linear, and quadratic terms). To compute these polynomials p_i, we use polynomial interpolation, *i.e.*, we need the output of these polynomials for several inputs. To do so, we exploit that the private key $K = (S, P, T)$ yields the same values as the public key. Therefore, we evaluate the function $T(P(S(x)))$ for several values of x:

- $\eta_0 \in \mathbb{F}^n$ is the 0 vector;
- $\eta_j \in \mathbb{F}^n : 1 \leq j \leq n$, is a vector with its j^{th} coefficient 1, the others 0;
- $\eta_{j,k} \in \mathbb{F}^n : 1 \leq j < k \leq n$, is a vector with its j^{th} and k^{th} coefficient 1 , the others 0.

These $1 + n + n(n-1)/2 = n(n+1)/2 + 1$ vectors yield the required coefficients, as we see below:

$$\begin{array}{rcl}
T(P(S(\eta_0)))_i &=& \alpha_i \\
T(P(S(\eta_j)))_i &=& \alpha_i + \beta_{i,j} + \gamma_{i,j,j} \\
T(P(S(a\eta_j)))_i &=& \alpha_i + a\beta_{i,j} + a^2\gamma_{i,j,j} \text{ where } a \in \mathbb{F}, a \neq 0, 1 \\
T(P(S(\eta_{j,k})))_i &=& \alpha_i + \beta_{i,j} + \beta_{i,k} + \gamma_{i,j,j} + \gamma_{i,k,k} + \gamma_{i,j,k} \,.
\end{array}$$

The values for $\alpha_i, \beta_{i,j}, \gamma_{i,j,k}$ are obtained by

$$\begin{array}{rcl}
\alpha_i &:=& T(P(S(\eta_0)))_i \\
\gamma_{i,j,j} &:=& \dfrac{1}{a(a-1)}[(T(P(S(a\eta_j)))_i - aT(P(S(\eta_j)))_i + (1-a)\alpha_i] \\
\beta_{i,j} &:=& (T(P(S(\eta_j)))_i - \gamma_{i,j,j} - \alpha_i \\
\gamma_{i,j,k} &:=& (T(P(S(\eta_{j,k})))_i - \gamma_{i,j,j} - \gamma_{i,k,k} - \beta_{i,j} - \beta_{i,k} + \alpha_i \,.
\end{array}$$

This yields the public polynomials $p_i(x_1, \ldots, x_n)$ for $1 \leq i \leq m$ in the case $\mathbb{F} \neq \mathrm{GF}(2)$.

To adapt the algorithm to $\mathbb{F} = \mathrm{GF}(2)$, we observe that $x^2 = x$ over GF(2), *i.e.*, all squares in only one variable become linear factors instead. Therefore, we can skip all terms with $\gamma_{i,j,j}$, *i.e.*, all quadratic terms in x_j^2 for $1 \leq j \leq n$. We can also take another point of view: as there is no element $a \in \mathrm{GF}(2) : a \neq 0, 1$, we could not evaluate $T(P(S(a\eta_j)))_i$ for such an a anyway.

2.2 Variations

In the previous section, we noted how HFE can be used for the encryption of messages and for signature generation. We now move on to the description of two important variations of HFE, namely HFE- and HFEv.

2.2.1 HFE-: Hiding Public Equations

Especially for signature schemes, an obvious change is to keep $1 < f < n$ polynomials $p_{n-f+1}, \ldots, p_n$ of the public key secret. As we discussed in Sect. 2.1.2, this is a necessity if the private polynomial P is not a surjection. However, even if the private polynomial P is a bijection, this might be a good idea as it keeps parts of the structure of the private key secret.

As for a signature scheme, keeping some polynomials $p_{n-f+1}, \ldots, p_n$ secret, is also expected to enhance the overall security of an encryption scheme. However, the number of equations removed can not be too high in this case. Indeed: keeping one equation p_n secret effectively means to take $\log_2 q$ bits of information out of $M' := \mathrm{HFE}(M)$. To restore these missing $\log_2 q$ bits, it is necessary to try all q possibilities for the encrypted messages $M'_1, \ldots, M'_q$ until the correct one is found. As the equation $P(x) = y$ has up to d solutions (see Sect. 2.1.1), we need to transmit some redundancy r anyway. However, to additionally compensate the loss of $\log_2 q$ bits of information, we will need to transmit more redundancy r. In addition, we need to solve up to q times the equation $P(x) = y$ for different values of y. As this is the most time consuming operation in HFE, it slows down the decryption process significantly. In general, by keeping f equations secret, we loose $\log_2 q^f$ bits of information and have to try up to q^f different possible encrypted messages $M'_1, \ldots, M'_{q^f}$, so we expect decryption to be $O(q^f)$ times slower. However, in terms of an attack, this modification is very difficult to overcome as we will see in Sect. 4

2.2.2 HFEv: Adding Vinegar Variables

While HFE- changes the public key, adding vinegar variables, *i.e.*, HFEv, changes the structure of the private polynomial P. Instead of using one private polynomial P, this modification allows q^v many private polynomials $P_1, \ldots, P_{q^v}$ where $v \in \mathbb{N}$ denotes the number of vinegar variables $z_1, \ldots, z_v$. As the private key should still be expressible in terms of at most quadratic polynomials $p_1, \ldots, p_n$, there is a restriction on the way these

q^v many private polynomials are obtained. In essence, the quadratic coefficients (*i.e.*, coefficients with a power of the form $q^a + q^b$ for some $a, b \in \mathbb{N}$) have to be the same for all these polynomials, while the linear coefficients depend on these vinegar variables $z_1, \ldots, z_v$ in an affine way, and the constant term depends on them in an at most quadratic way. In symbols:

$$\begin{aligned} P_{(z_1,\ldots,z_v)}(x) \quad := \quad & \sum_{\substack{0 \le i,j \le d \\ q^i+q^j \le d}} a_{i,j} x^{q^i+q^j} + \sum_{\substack{0 \le k \le d \\ q^k \le d}} b_k(z_1, \ldots, z_v) x^{q^k} + c(z_1, \ldots, z_v) \\ & \text{for } a_{i,j} \in \mathbb{E}, \\ & b_k(z_1, \ldots, z_v) \text{ are affine in } (z_1, \ldots, z_v), \text{ and} \\ & c(z_1, \ldots, z_v) \text{ is at most quadratic in } (z_1, \ldots, z_v) \end{aligned}$$

For a signature scheme, HFEv can be implemented very easily. The vinegar variables $(z_1, \ldots, z_v) \in_R \mathbb{F}^v$ are initialised with random values. After this step, there is only one private polynomial P, so the rest of the algorithm keeps unchanged. For an encryption scheme, it is not so easy to introduce this v modification as a priori any of the q^v possible settings for the vinegar variables is equally likely. As in HFE-, it is necessary to check up to q^v different equations $P(x) = y$ — but in this case, not the value of y but the polynomial itself changes. So for a signature scheme, it is possible to have many vinegar variables, while in an encryption scheme, their number must be small.

2.2.3 Working with a Subfield $\widetilde{\mathbb{F}}$ of $\mathbb{F}$

While the aim in the last section was to enhance security, we now want to concentrate on the size of the public key. To obtain a smaller public key, we can choose both the coefficients in the two affine transformations S and T and also the coefficients in the private polynomial P in a way that it is possible to express the public key $k = (p_1, \ldots, p_n)$ in terms of a proper subfield of $\mathbb{F}$. For example, for $\mathbb{F} = GF(128)$ and the subfield $\widetilde{\mathbb{F}} = GF(2)$, we save $\frac{6}{7}$ of the size of the public key as each coefficient in the public polynomials $(p_1, \ldots, p_n)$ does no longer require 7 bits, but only 1 bit. However, the message space is still $\mathbb{F}^n = GF(128^n)$ over the original finite field GF(128).

Unfortunately, this modification has two drawbacks. First of all, it leads to a big degree d, as $d = q^a + q^b$ for some $a, b \in \mathbb{N}$ grows exponentially with $q = |\mathbb{F}|$. Secondly, it reduces dramatically the key space for the private key. As [Pat96] points out, it is possible that an attacker could make use of this special structure of the public key. In fact, this was the case for Sflash^{v1} [CGP00b] — which is a C* scheme and uses $\mathbb{F} = GF(128)$, $\widetilde{\mathbb{F}} = GF(2)$ and $n = 37$. Gilbert and Minier [GM02] observed that we are able to express the *whole* Sflash system over $\widetilde{\mathbb{F}} = GF(2)$. By the field properties, all vectors in the message space with coefficients from $\widetilde{\mathbb{F}}$ will be mapped to vectors over $\widetilde{\mathbb{F}}$ in the signature space. This way, it is possible to perform a brute-force search on a total of 2^{37} points — which is certainly much less than the claimed security level of 2^{80}. Further exploiting the fact that Sflash is a bijection, they are able to recover the private key. The attack complexity is about $|\widetilde{\mathbb{F}}|^n$, *i.e.*, 2^{37} here.

A key point in their attack is the fact that Sflash has a bijective structure. As we saw in sections 2.1.1 and 2.1.2, this is not the case for HFE. Therefore, using coefficients from a subfield may not be dangerous for HFE. However, there are two concerns. First, having a ground field $\mathbb{F} \neq \mathrm{GF}(2)$ leads to a rather large degree as $d = O(|\mathbb{F}|^a)$ for $a \in \mathbb{N}$. This leads to rather slow decryption / signature generation (see above). Second, we will obtain much less coefficients on the private polynomial P if we use a ground field $\mathbb{F} \neq \mathrm{GF}(2)$. In the light of the recent attacks of Faugère and Joux (cf Sect. 4.2), this is a dangerous situation. Therefore we conclude that this modification may be save for HFE, but it is certainly not of practical interest.

2.2.4 Modifications Revisited

It follows from this section that HFE is a very flexible scheme which allows many modifications. Due to space limitations in this paper, we were not able to describe all modifications for HFE known so far, *e.g.*, HFE+ or more than one branch. They can be found in [Pat96] (cf [Wol02] for a more detailed explanation). Thus HFE can be adapted to different application domains and also react on different attacks by slightly changing its overall structure. In the remainder of this section we go briefly through these different variations on HFE and determine their use for encryption and signature.

For a signature scheme, HFE- and HFEv are certainly very useful as they do not slow down signature generation and enhance the overall security of HFE. In contrast, both HFE with more branches and HFE using a subfield of $\mathbb{F}$ for the public key, seem to lead to a less secure scheme. On the other hand, both modifications change HFE in a very desirable way: the first leads to a speed-up for signature generation while the second yields a smaller public key. All in all, combining the v modification with HFE- can be used to construct both fast and secure digital signature schemes (see Sect. 3).

For encryption, the situation seems to be worse. Both HFE- and HFEv will lead to a rather slow decryption process so neither too many equations can be removed (HFE-) nor too many variables added (HFEv). For other modifications, such as HFE+, the situation is better. In this case, the decryption step takes the same amount of time, however, the public key has more equations than variables. As solving over-defined equations in many variables looks sub-exponential (cf [Cou01]), it seems to be a good idea to have nearly the same number of equations as variables. A special choice for the private polynomial P, *e.g.*, a bijection, is not secure [Pat96, Sect. 7.3]. And combining this with HFE- does not yield results either, as we cannot remove too many equations from an encryption scheme. Having two or more branches for the private polynomial P or using a subfield of $\mathbb{F}$ seems to have no special effect for an encryption scheme, so the same caution is necessary as for a signature scheme.

3 QUARTZ

In Sect. 2.2, we looked at two important modifications of HFE, namely HFE- and HFEv. In this section, we will see how they can be combined to obtain a practical signature scheme, namely Quartz. It was submitted to NESSIE [NES] but was not accepted. The purpose of this section is to describe why it failed.

3.1 Historical Note

The design goal of Quartz was not only to withstand all known attacks but also to have good chances to withstand future attacks as well. So the parameters in Quartz have been chosen rather conservatively, which results in a rather long signature time, namely 10s on average on a Pentium II 500 MHz [CGP01]. As we know now (March 2004), the choice of parameters was not conservative enough. We will discuss this point in more detail in Sect. 4. When not stated otherwise, this section is based on [CGP01] and describes the second, reversed version of Quartz. The changes made from the first version (cf [CGP00a]) to the second version of Quartz are not due to security problems. On the one hand, they were made to speed up the whole algorithm without jeopardising its security. On the other hand, they allow a security proof for Quartz.

3.2 System Parameters

Table 1: Parameter for Quartz

Parameter	Quartz
$q = \|\mathbb{F}\|$	2
$n = \partial i(t)$	103
transformation S	$\mathbb{F}^{107} \rightarrow \mathbb{F}^{107}$
transformation T	$\mathbb{F}^{103} \rightarrow \mathbb{F}^{103}$
l (equations removed)	3
v (vinegar variables)	4
m (equations)	100
n (variables)	107
d (degree)	129
Signature Length	128 bits
Private Key Size	3 kB
Public Key Size	71 kB

As we see in Table 1, the signature length (128 bits) is 21 bits larger than expected: as the extension field $\mathbb{E}$ has dimension 103 while 4 vinegar variables are added, we would expect a signature length of 107 bits. The reason for this difference lies in the fact that Quartz uses a so-called "Feistel-Patarin-Network" (sometimes also denoted "Patarin-Chained-Construction") to compute the signature. Within this network, the HFE algorithm is called four times to compute a signature, *i.e.*, this involves solving the underlying HFE problem four times (cf Tabel 6 and Sect. 3.3.2). This way, we need to add 4 times 7 bits to the number of public equations.

3.3 System Description

To deal with the different security features of Quartz, we have to look at them and try to deduce if they enhance or jeopardise the security of Quartz. First of all, the private polynomial P has full coefficients, *i.e.*, it has non-trivial coefficients from $\mathbb{E}$ and also all possible coefficients, *i.e.*, every power which has Hamming weight two or lower. Together with the vinegar variables (denoted $z_1, \ldots, z_4$), the private polynomial P of Quartz can be expressed as:

$$\begin{aligned} P_{(z_1,\ldots,z_4)}(x) \quad := \quad & \sum_{\substack{0 \le i,j \le 7 \\ q^i+q^j \le 129}} a_{i,j} x^{q^i+q^j} + \sum_{\substack{0 \le k \le 7 \\ q^k \le 129}} b_k(z_1,\ldots,z_v) x^{q^k} + c(z_1,\ldots,z_v) \\ & \text{for } a_{i,j} \in \mathbb{E}, \\ & b_k(z_1,\ldots,z_v) \text{ are affine in } (z_1,\ldots,z_v), \text{ and} \\ & c(z_1,\ldots,z_v) \text{ is at most quadratic in } (z_1,\ldots,z_v) \end{aligned}$$

As the polynomial has all coefficients and the degree is rather high, Quartz withstood at design time all known attacks up to a complexity level of 2^{80} — this level was requested for signature algorithms in NESSIE. This is also true if there is no v modification. In fact, the degree is very high as in 2000 a degree of 25–33 was estimated to be high enough. In addition, Quartz is a HFEv rather than a "basic" HFE scheme. This modification is expected to further enhance the security of Quartz. Moreover, Quartz is also a HFE-scheme with 3 equations kept secret. As Quartz uses a very general polynomial P and also the v modification, the attacks known against basic HFE do not apply against Quartz. So removing only three equations from the public key seems to be sufficient for Quartz and is expected to enhance its overall security. We call the parts of Quartz discussed above the "HFE"-step, *i.e.*, $\text{HFE}(x) := T(P(S(x)))$ and its inverse $\text{HFE}^{-1}(y) := S^{-1}(P^{-1}(T^{-1}(y)))$.

3.3.1 Precomputations and HFE-step

Although the HFE-step itself looks quite secure, there is an obvious attack using the birthday paradox: by computing 2^{50} different versions of the message M and by applying the public key to 2^{50} different values for $x_1, \ldots, x_n$ we expect to obtain a valid signature for one version of the message M as the HFE step alone uses only 100 public equations. This is far less than the complexity level of 2^{80} required in NESSIE. To overcome this

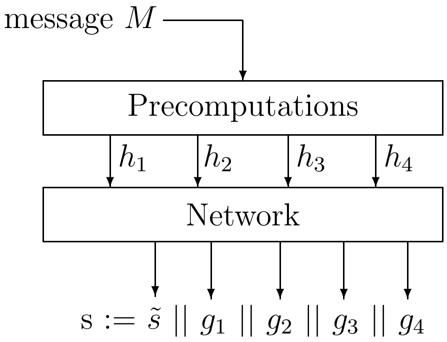

Fig. 4: Overall Structure of Quartz for Signature Generation

problem, Quartz combines four invocations of the HFE-step in a so-called "Feistel-Patarin network". The key idea of this network is not to store four times a full signature (*i.e.*, a signature of 400 bits in total) but to save only the last signature completely. In addition, it stores 7 bits for each of the 4 signatures computed. The reason for this lies in the fact that the HFE-step of Quartz has only 100 bits of input but a 107 bit output. These additional 7 bits compensate for this expansion. The overall structure of Quartz is shown in Fig. 4. As we see there, signature generation with Quartz requires a precomputation step (see

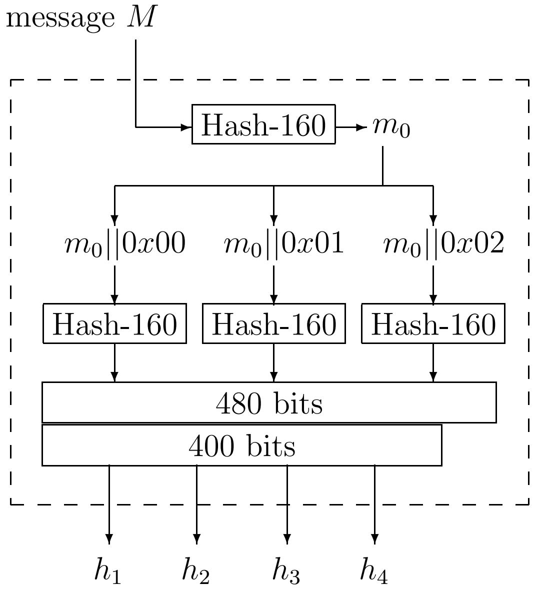

Fig. 5: Precomputation in Quartz

Fig. 5) before applying the network itself (see Fig. 6). The key idea of the precomputation step is to use three calls of a 160-bit hash function (SHA-1 in Quartz, cf [FIP] for SHA-1) to "expand" a 160-bit hash (denoted m_0 in Fig. 5) to four 100-bit values h_1, h_2, h_3 and h_4. During this process, the original hash m_0 is concatenated (operator $\cdot||\cdot$) with the 8 bit values $0x00$, $0x01$ and $0x02$ (C notation for the numbers 0, 1 and 2) to obtain three

168 bit values. Each of them is hashed individually using a 160 bit hash function and then concatenated. The resulting 480 bit number is truncated to 400 bits and yields four 100-bit strings. If Quartz used a hash function with a 512 bit output rather than 160 bit, the precomputation step would be obsolete. Such functions have been accepted in the NESSIE project (*e.g.*, algorithm "Whirlpool") and also in NIST (only the algorithm in [FIP01]). But for the complexity level of 2^{80}, it is sufficient to use a 160 bit hash function and to expand its output to 400 bits as done in the precomputation step of Quartz.

3.3.2 Feistel-Patarin network

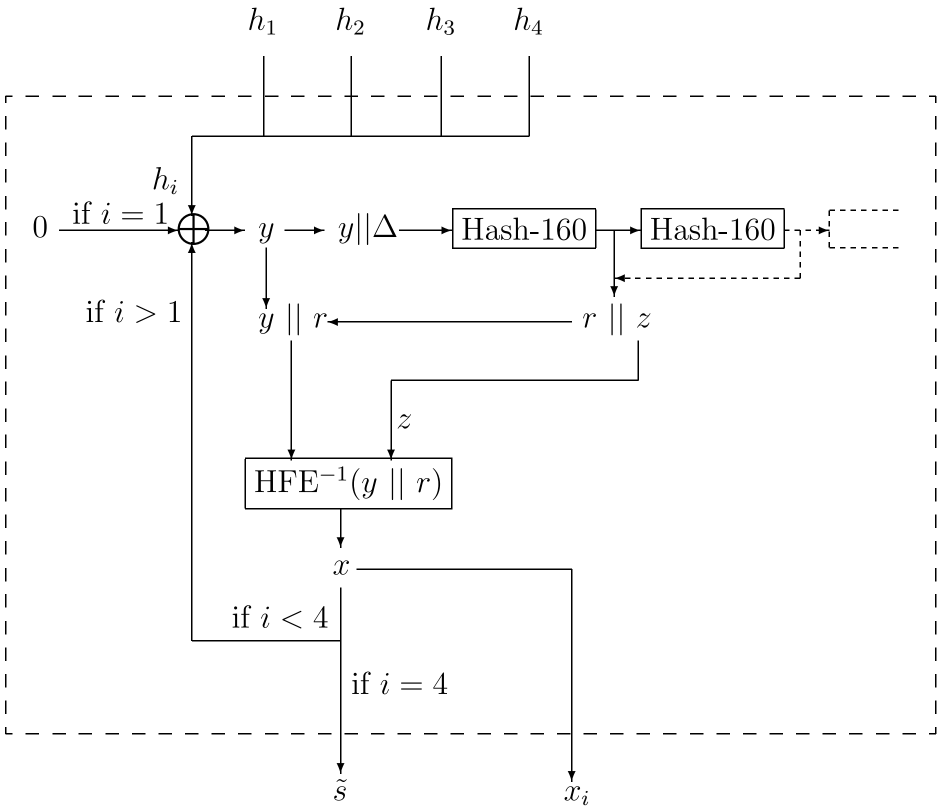

Fig. 6: Central Structure of the Feistel-Patarin network for Quartz

We will now concentrate on the "Feistel-Patarin network" itself as outlined in Fig. 6. It uses the output of the precomputation step as input. We describe the first step of the network. After initialising the counter i with 0, it "xors" h_0 with 0 to obtain the intermediate value y. This is certainly obsolete as h_0 "xor" $0 = h_0$. However, during the run of the algorithm, h_1, h_2, h_3 are "xored" with the output of the previous step, so this "xor" operation is required for symmetry of the four steps. After this initialisation, the 100-bit value y is hashed together with a secret 80 bit parameter Δ to obtain the random variables r (3 bit) and the vinegar variables z (4 bit). Both are fed into the HFE step to obtain a valid signature of 107 bits. According to [CGP01, Sec. 5.3] and as previously noted in Sect. 2.1.2 of this paper, the probability to obtain a valid signature at one attempt is $\approx 60\%$. If there is no valid signature, the hash of $y \mid\mid \Delta$ is rehashed. This is repeated until a valid signature is obtained. The probability that there is no valid signature at all for a given message M is estimated to be $\leq 2^{-183}$ and hence negligible [CGP01, Sec. 5.3]. If a valid 107 bit signature is found, the least significant 100 bits of x are fed back into the network while the most significant 7 bits are stored as output g_1. The other three steps are similar but h_i is not "xored" with 0 but with the least significant 100 bits of x. In the final step, these 100 bits are not fed into the network but yield output $\tilde{s}$. In each step, it is possible that there is not only one, but up to $d = 129$ different solutions for the equation $x = HFE(y \mid\mid r)$. The Quartz-specification states that only one is chosen, namely the one with the least hash value (bit-wise comparison without sign bit).

3.3.3 Signature Verification

To verify the validity of a signature, this network is reversed. As the public key consists of 100 polynomials $p_1, \ldots, p_{100}$ in 107 input variables $x_1, \ldots, x_{107}$, the 7 bit values $g_1, \ldots, g_4$ are used to obtain 107 bits input for the public key during each run. In addition, as the four 100 bit values $h_1, \ldots, h_4$ are "xored" each time, a signature is only valid if the overall output of this scheme is 0. In this case the signature is accepted.

3.4 Conclusions

The Feistel-Patarin network is certainly a rather complicated security feature. However, as each signature depends on a 400 bit input (which is obtained from a 160 bit hash value), it seemed to be a rather strong signature system. Moreover, as Quartz uses the 160-bit hash function as a kind of cryptographically secure random number generator, it is deterministic, so each message has always the same signature (for the same private key K). In the original specification of HFE as a signature scheme it seemed to be necessary to use "real" randomness to obtain valid signatures. As real randomness often is a problem (*e.g.*, in a stand-alone server without user interaction), this deterministic version makes it possible to use Quartz in more application domains.

As we will see in the attack section, Quartz as proposed in [CGP01] is no longer considered to be secure. Consequently, it has not been recommended by NESSIE [PBO+03].

4 ATTACKS

In this section, we give a brief overview of recent attacks against HFE. Due to space limitations, we can only sketch the corresponding articles. For a more detailed but partly outdated analysis, we refer to [Pat96].

4.1 Kipnis-Shamir: Recover the Private Key

In [KS99], Kipnis and Shamir show how to recover the private key of HFE from the system of public equations. The key point of this attack is to express the private key (*i.e.*, polynomials over the finite field $\mathbb{F}$) as sparse univariate polynomials over the extension field $\mathbb{E}$. In addition, they observe that the special choice of the private polynomial P in HFE gives rise to a matrix equation with very small rank (*e.g.*, rank 13 for a 100×100 matrix). In [Cou01, Sect. 8], their attack is improved and has now a workload of $\binom{n}{\mathrm{Rank}P}^{\omega} = \mathcal{O}\left(n^{\log_q d + \mathcal{O}(1)}\right)$. In this formula, Rank$P$ is the rank of the private polynomial P in matrix form over the extension field $\mathbb{E}$, d its degree as a polynomial over $\mathbb{E}$, and $\omega \approx 2.7$ the workload of solving linear equations. The attack is only applicable against basic HFE, *i.e.*, fails for all its variations. On the other hand, it is the only attack known so far which can recover the private key of HFE.

In their paper, Kipnis and Shamir also introduce the "reliniarization" technique which can solve quadratic equations with about $0.1n^2$ linearly independent equations in n variables. For the traditional linearization technique, we need about $0.5n^2$ many equations. This technique has been improved in [CKPS00].

4.2 Faugère: Fast Gröbner Bases

In 2002, Faugère reported to have broken the HFE-Challenge I in 96 hours [Fau02]. Since then, his attacks have been improved and in 2003, Faugère and Joux published joint work on the security of HFE [FJ03] (cf [Fau03] for a more technical version). In a nutshell, their attack uses a fast algorithm to compute the Gröbner basis of a system of polynomial equations. By theoretical and empirical studies they show that basic HFE is polynomial for a fixed degree. The attack-complexity for different degrees is shown in Table 2

Table 2: Attack complexity against basic HFE for different degrees d

Degree d	$16 < d \leq 128$	$128 < d \leq 512$	$512 > d$
Attack (asymptotical)	$\mathcal{O}(n^8)$	$\mathcal{O}(n^{10})$	$\geq \mathcal{O}(n^{12})$
Attack (for $n = 103$)	$\approx 2^{54}$	$\approx 2^{66}$	$\approx 2^{80}$

For HFEv, Faugère and Joux outline in [FJ03, Sect. 4.1] that the cryptanalysis is not more difficult in this case. But for HFE-, they get a higher workload. For the original Quartz-scheme, they establish a workload of $\approx 2^{62}$ — exploiting some further properties of their attack. However, these additional improvements are not within the scope of this paper.

Using the estimations of [FJ03, Sect. 4.1, 5.2–5.4] on Quartz, we establish that a degree of 129 and 7 equations removed (thus, without the modification HFEv) has an attack complexity of $\approx 2^{86}$. The corresponding "Quartz-7m"-scheme is therefore secure again. In fact, a similar result has been achieved 2002 in [CDF03] by increasing the degree d of the private polynomial to 257. However, this estimation was only based on [Fau02]. In the light of the article [FJ03] it turns out to be inaccurate.

4.3 Secure Versions of Quartz

At present, we see two possibilities to obtain a secure version of the Quartz signature algorithm which is able to withstand the attack from [FJ03]. The first uses a degree of 513 for the public polynomial and keeps the other parameters unchanged. We call this version Quartz-513d and expect an attack complexity of $\approx 2^{82}$. However, due to the very large degree of the private polynomial, we do not expect this version to be of practical interest.

Table 3: Parameter for different versions of Quartz

Parameter	Quartz	Quartz-7m	Quartz-513d
$q = \vert\mathbb{F}\vert$	2		
$\partial i(t)$ (degree $\mathbb{E}$)	103	107	103
transformation S	$\mathbb{F}^{107} \to \mathbb{F}^{107}$		
transformation T	$\mathbb{F}^{103} \to \mathbb{F}^{103}$	$\mathbb{F}^{107} \to \mathbb{F}^{107}$	$\mathbb{F}^{103} \to \mathbb{F}^{103}$
l (equations removed)	3	7	3
v (vinegar variables)	4	0	4
m (equations)	100		
n (variables)	107		
d (degree)	129	129	513
Signature Length	128 bits		
Private Key Size	3 kB	3kB	4kB
Public Key Size	71 kB		
Security Level	2^{62}	2^{82}	2^{86}

Therefore, we concentrate on the version already outlined in the previous section: replace the 4 vinegar variables by removing 4 equations. The corresponding system has still the same signature size as Quartz but an estimated attack complexity of $\approx 2^{86}$. It therefore meets the NESSIE-requirements of 2^{80} TDES-computations.

Although these versions (cf Table 3) are secure against the recent attack from Faugère and Joux, we argue to be cautious as they have not been independently studied by other researchers. It is therefore well possible that they carry unnoticed weaknesses.

5 CONCLUSIONS

In this paper, we outlined the structure of the Hidden Field Equations system (HFE) from Patarin and described two important variations, namely HFE- and HFEv. Using these variations, we described the original Quartz signature scheme. In the light of recent attacks, this scheme can no longer be considered to be secure. Therefore, we outlined how the internal structure of Quartz (*i.e.*, its private key) can be changed in order to counter these attacks. In particular, neither the rather short signature size of 128 bits nor the signature-verification process of Quartz is affected by our proposed changes (cf Table 3).

This shows that Hidden Field Equations are still an interesting research topic which allow exceptional short signature sizes. However, before using HFE in practice, further research is needed to fully understand its security.

Acknowledgements

We want to thank Patrick Fitzpatrick and Simon Foley (University College Cork, Ireland) for their comments on an earlier version of this paper. We also want to thank Kaisa Nyberg (Nokia, Finland), Jacques Patarin (University of Versailles, France), and Magnus Daum (University of Bochum, Germany) for helpful remarks. This work was supported in part by the Concerted Research Action (GOA) Mefisto-2000/06 of the Flemish Government.

REFERENCES

[BWP05] An Braeken, Christopher Wolf, and Bart Preneel. A study of the security of Unbalanced Oil and Vinegar signature schemes. In *The Cryptographer's Track at RSA Conference 2005*, Lecture Notes in Computer Science. Alfred J. Menezes, editor, Springer, 2005. 13 pages, cf `http://eprint.iacr.org/2004/222/`.

[CDF03] Nicolas T. Courtois, Magnus Daum, and Patrick Felke. On the security of HFE, HFEv- and Quartz. In *Public Key Cryptography — PKC 2003*, volume 2567 of *Lecture Notes in Computer Science*, pages 337–350. Y. Desmedt, editor, Springer, 2002. `http://eprint.iacr.org/2002/138`.

[CGP00a] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *Quartz: Primitive specification and supporting documentation*, 2000. `https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/quar%tz.zip`, 15 pages.

[CGP00b] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *SFlash: Primitive specification and supporting documentation*, 2000. `https://www.cosic.esat.kuleuven.ac.be/nessie/workshop/submissions/sfla%sh.zip`, 10 pages.

[CGP01] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *Quartz: Primitive specification (second revised version)*, October 2001. `https://www.cosic.esat.kuleuven.ac.be/nessie` Submissions, Quartz, 18 pages.

[CGP03] Nicolas Courtois, Louis Goubin, and Jacques Patarin. *SFlash^{v3}, a fast asymmetric signature scheme — Revised Specificatoin of SFlash, version 3.0*, October 17th 2003. ePrint Report 2003/211, `http://eprint.iacr.org/`, 14 pages.

[CKPS00] Nicolas T. Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Bart Preneel, editor, Springer, 2000. Extended Version: `http://www.minrank.org/xlfull.pdf`.

[Cou01] Nicolas T. Courtois. The security of Hidden Field Equations (HFE). In *The Cryptographer's Track at RSA Conference 2001*, volume 2020 of *Lecture Notes in Computer Science*, pages 266–281. D. Naccache, editor, Springer, 2001. `http://www.minrank.org/hfesec.{ps|dvi|pdf}`.

[Dau01] Magnus Daum. Das Kryptosystem HFE und quadratische Gleichungssysteme über endlichen Körpern. Diplomarbeit, Universität Dortmund, August 2001. `http://homepage.ruhr-uni-bochum.de/Magnus.Daum/HFE.{ps.zip|pdf}`, 133 pages.

[Fau02] Jean-Charles Faugère. HFE challenge 1 broken in 96 hours. Announcement that appeared in `news://sci.crypt`, 19th of April 2002.

[Fau03] Jean-Charles Faugère. Algebraic cryptanalysis of (HFE) using Gröbner bases. Technical report, Institut National de Recherche en Informatique et en Automatique, February 2003. `http://www.inria.fr/rrrt/rr-4738.html`, 19 pages.

[FIP] National Institute of Standards and Technology. *Federal Information Processing Standards Publication 180-1: Secure Hash Standard*, 17th April 1995. `http://www.itl.nist.gov/fipspubs/fip180-1.htm`.

[FIP01] National Institute of Standards and Technology. *Federal Information Processing Standards Publication 180-2: Secure Hash Standard*, August 2001. `http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf`.

[FJ03] Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of Hidden Field Equations (HFE) using gröbner bases. In *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 44–60. Dan Boneh, editor, Springer, 2003.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979. ISBN 0-7167-1044-7 or 0-7167-1045-5.

[GM02] Henri Gilbert and Marine Minier. Cryptanalysis of SFLASH. In *Advances in Cryptology — EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 288–298. Lars R. Knudsen, editor, Springer, 2002.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem. In *Advances in Cryptology — CRYPTO 1999*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Michael Wiener, editor, Springer, 1999. `http://www.minrank.org/hfesubreg.ps` or `http://citeseer.nj.nec.com/kipnis99cryptanalysis.html`.

[MI88] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature verification and message-encryption. In *Advances in Cryptology — EUROCRYPT 1988*, volume 330 of *Lecture Notes in Computer Science*, pages 419–545. Christoph G. Günther, editor, Springer, 1988.

[MvOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. ISBN 0-8493-8523-7, online-version: `http://www.cacr.math.uwaterloo.ca/hac/`.

[NES] NESSIE: New European Schemes for Signatures, Integrity, and Encryption. Information Society Technologies programme of the European commission (IST-1999-12324). `http://www.cryptonessie.org/`.

[Pat95] Jacques Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In *Advances in Cryptology — CRYPTO 1995*, volume 963 of *Lecture Notes in Computer Science*, pages 248–261. Don Coppersmith, editor, Springer, 1995.

[Pat96] Jacques Patarin. Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of asymmetric algorithms. In *Advances in Cryptology — EUROCRYPT 1996*, volume 1070 of *Lecture Notes in Computer Science*, pages 33–48. Ueli Maurer, editor, Springer, 1996. Extended Version: `http://www.minrank.org/hfe.pdf`.

[PBO+03] B. Preneel, A. Biryukov, E. Oswald, B. Van Rompay, L. Granboulan, E. Dottax, S. Murphy, A. Dent, J. White, M. Dichtl, S. Pyka, M. Schenfheutle, P. Serf, E. Biham, E. Barkan, O. Dunkelman, J.-J. Quisquater, M. Ciet, F. Sica, L. Knudsen, M. Parker, and H. Raddum. NESSIE security report, version 2.0. Document NES/DOC/ENS/WP5/D20/2, 19th of February 2003. `https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D20-v2.pdf`, see [NES], 342 pages.

[PG97] Jacques Patarin and Louis Goubin. Trapdoor one-way permutations and multivariate polynomials. In *International Conference on Information Security and Cryptology 1997*, volume 1334 of *Lecture Notes in Computer Science*, pages 356–368. International Communications and Information Security Association, Springer, 1997. Extended Version: `http://citeseer.nj.nec.com/patarin97trapdoor.html`.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[WBP04] Christopher Wolf, An Braeken, and Bart Preneel. Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC. In *Conference on Security in Communication Networks — SCN 2004*, Lecture Notes in Computer Science, pages 145–151, September 8–10 2004. Extended version: `http://eprint.iacr.org/2004/237`.

[Wol02] Christopher Wolf. *Hidden Field Equations* (HFE) - variations and attacks. Diplomarbeit, Universität Ulm, December 2002. `http://www.christopher-wolf.de/dpl`, 87 pages.

[Wol03] Christopher Wolf. Efficient public key generation for multivariate cryptosystems. Cryptology ePrint Archive, Report 2003/089, 5th of May 2003. `http://eprint.iacr.org/2003/089/`, 12 pages.

[Wol04] Christopher Wolf. Efficient public key generation for HFE and variations. In *Cryptographic Algorithms and Their Uses 2004*, pages 78–93. Dawson, Klimm, editors, QUT University, 2004.

[WP04] Christopher Wolf and Bart Preneel. Asymmetric cryptography: Hidden Field Equations. In *European Congress on Computational Methods in Applied Sciences and Engineering 2004*. P. Neittaanmäki, T. Rossi, S. Korotov, E. Oñate, J. Périaux, and D. Knörzer, editors, Jyväskylä University, 2004. 20 pages, extended version: `http://eprint.iacr.org/2004/072/`.

[WP05a] Christopher Wolf and Bart Preneel. Equivalent keys in HFE, C*, and variations. In *Proceedings of Mycrypt 2005*, volume 3715 of *Lecture Notes in Computer Science*, pages 33–49. Serge Vaudenay, editor, Springer, 2005. Extended version `http://eprint.iacr.org/2004/360/`, 12 pages.

[WP05b] Christopher Wolf and Bart Preneel. Superfluous keys in $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic asymmetric systems. In *Public Key Cryptography — PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 275–287. Serge Vaudenay, editor, Springer, 2005. Extended version `http://eprint.iacr.org/2004/361/`.

[WP05c] Christopher Wolf and Bart Preneel. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint Archive, Report 2005/077, 12th of May 2005. `http://eprint.iacr.org/2005/077/`, 64 pages.