Multivariate Encryptions with LL' perturbations - Is it possible to repair HFE in encryption? -

Jacques Patarin¹² and Pierre Varjabedian¹

 ¹ THALES, Meudon, France,
 ² Université Paris-Saclay, UVSQ, CNRS, Laboratoire de mathématiques de Versailles, 78000, Versailles, France, {jacques.patarin,pierre.varjabedian}@thalesgroup.com

Abstract. We will present here new multivariate encryption algorithms. This is interesting since few multivariate encryption scheme currently exist, while their exist many more multivariate signature schemes. Our algorithms will combine several ideas, in particular the idea of the LL' perturbation originally introduced, but only for signature, in [GP06]. In this paper, the LL' perturbation will be used for encryption and will greatly differ from [GP06]. As we will see, our algorithms resists to all known attacks (in particular Gröbner attacks and MinRank attacks) and have reasonable computation time.

Keywords: Post-Quantum public key encryption · Post-Quantum key exchange · Multivariate cryptography · HFE

1 Introduction

Multivariate Cryptography is one of the main families of algorithms used to make post-quantum public key cryptography. We can trace it's origin with the article [MI88] that introduced the so called C* scheme. The scheme was broken in [Pat95] but it lead to multiple algorithms. With multivariate cryptography it is supposedly possible to do encryption, signatures and authentication. Yet, these last years almost all know multivariate encryption algorithms have been broken due to the discovery and improvements of attacks such as the MinRank attacks. Due to this, at present most of the algorithms that are basing their security on the MQ problem can only be used for signatures or authentications. (For example the UOV [KPG99] family is currently largely represented in the recent NIST standardisation process).

Another major multivariate algorithm is Hidden Field Equations (HFE) [Pat96]. Until recently HFE was one the main candidate for multivariate cryptography. Especially since it was one of the few able to be use for encryption (contrary to UOV previously mentioned) as well as signatures. Since it's origin it was suggested to add perturbations in order to reinforce it's security.

HFE has a lot of variants. In the recent article [CMRPV24] the most important perturbations are presented with their current cryptanalysis properties. In this article, the authors showed that it is hard to use HFE for encryption, indeed the perturbations that they studied have a high cost in order to avoid all attacks. However, the authors of [CMRPV24] have proposed a variant that seems to work for signatures. Other tries around HFE or more generally around multivariate cryptography in encryption have been made [SDP16], [IPS⁺18], [BM09], [CBD⁺09]. They generally have failed to resist to all of the attacks [CCFST24], [BM09] or have been found to be far less effective than expected. This is why it is considered to be much easier to make multivariate signature schemes than encryption scheme.

The main novelty of this article is that we will be using the perturbation LL' introduced in [GP06] in a specific and original way for encryption. Our LL' can also be viewed as a particular case of the other perturbation Hat+ [FmRPP22], but as we will see, our LL' is much more efficient than Hat+.

2 Vanilla HFE

Here by "vanilla" HFE or "Nude" HFE we mean the basic HFE scheme, *i.e.* without any additional perturbation.

The main idea for algorithm based on the difficulty to solve a quadratic system (Multivariate Quadratic problem) is to find structured quadratic polynomial that makes the problem easy to solve. Then we will hide their structure via secret linear transform. For example let f a quadratic polynomial such that the equation in x, f(x) = y is easy to solve. Such polynomial f will be called the central map of the system. Let T, S be two linear transform. The public key will be $T \circ f \circ S$, that way the public key will hopefully mimic the behaviour of a random quadratic map. Here the private key will be: T, S, f, hence with the private key and the hypothesis that f is easily "invertible" we can make an encryption/signature algorithm.

HFE uses such structure. HFE use the fact that \mathbb{F}_{q^n} can be seen as the vector space \mathbb{F}_q^n with a canonical morphism. Then an univariate polynomial f(x) of \mathbb{F}_{q^n} can also be seen as a multivariate polynomial $f(x_1, \ldots, x_n)$. For vanilla HFE f(x) will be a polynomial like this:

$$f(x) = \sum_{i,j \le d} a_{ij} x^{q^i + q^j}$$

We see that the monomials are quadratic in \mathbb{F}_q and that the degree is limited by $D = 2 * q^d$. This is to ensure that Berlekamp algorithm will find a solution in a reasonable time. Then let T, S be two linear transform $\mathbb{F}_q^n \to \mathbb{F}_q^n$. Let $\phi : \mathbb{F}_{q^n} \to \mathbb{F}_q^n$ be the canonical morphism from the field extension of \mathbb{F}_q to the vector space. Then the public key will be :

$$T \circ \phi \circ f \circ \phi^{-1} \circ S.$$

However most of the time in literature the morphism ϕ will not be written but only implied. So the public key could also be written as :

$$T \circ f \circ S.$$

The private key will be :

T, S, f.

From an attacker point of view the goal would either be to invert the system directly (via Gröbner basis for example) or recover the linear transform T or S(via a MinRank attack for example). Both will be using the fact that the degree of the central map f is limited.

N.B. When the level of security is higher than 80bits, vanilla HFE require large D, therefore unrealistic computation time. This is why we need perturbations from the main algorithm in order to achieve higher level of security.

LL' perturbation 3

3.1Our LL' perturbation in encryption

Let P_1, \ldots, P_n be the public key of a multivariate central map (e.g HFE). P_1, \ldots, P_n are polynomials of degree 2 in x_1, \ldots, x_n . Let r be an integer such that $1 \leq r \leq n$. Let L_i , $1 \leq i \leq r$ and L'_i , $1 \leq i \leq r$ be 2 r secret linear forms in $x_1, \ldots x_n$. Our "outside" LL' consist in adding secret linear combination $\sum_{i} \alpha_{i,j} L_i L'_i$ to the equations P_j , $1 \le j \le n$ therefore $P'_j = P_j + \sum_{i} \alpha_{i,j} L_i L'_i$

$\mathbf{3.2}$ Analog description

Let A_1, \ldots, A_n the secret analog equations of this central map $(A_1, \ldots, A_n$ are secret polynomials of degree 2 in a_1, \ldots, a_n . In fact we can go from A_i to P_i via the linear transformation S and T. Our "inside" LL' consist in adding secret linear combinations to the A_j , $1 \le j \le n$ so $A'_j = A_j + \sum_i \alpha_{i,j} L_i L'_i$. In fact our "inside" and "outside" transformation are similar (since we can go from one to the other description by just modifying S and T), but for the analysis we will sometime alternatively use "inside" or "outside".

Variants 3.3

- Usually, x_1, \ldots, x_n will be on the small field GF(2). Yet, this perturbation LL' can work on other very small Field (GF(3) for example) however it will be less efficient as the probability for a linear form to be null on a point will decrease.
- Usually, we will have public equations of degree 2. Later in this paper we will also present a variant with equations of degree 3.
- It is also possible to use very simple linear combination for $L_i L'_i$. Instead of a general linear combination of the form: $\sum_i \alpha_{i,j} L_i L'_i$ we can imagine to just add $L_i L'_i$ to P_i . At present, this variant does not seems to reduce the security, but it does not accelerate the decryption either.

- In this paper, we will use our LL' perturbation with HFE central map. Other central maps may also be used, for example C* ,cf. [MI88]. However, so far all attempts to repair C* have failed. Therefore we are more confident with HFE and a specific analysis will probably be required on C*.

Remark: A central map (as HFE for example) is really needed, *i.e.* LL' cannot be used alone. This is because a random linear combination of LL' equation will have a non-negligible probability to have a small rank and from this property it is possible to find \mathbf{S} or \mathbf{T}

3.4 Decryption

The main idea of the LL' perturbation is that on \mathbb{F}_2 we have a probability 1/4 that $L_i L'_i \neq 0$. In order to decrypt, we will evaluate the expected number of products $L_i L'_i \neq 0$ (it will be r/4 in average, but it can be much less as we will see below). Then, after exhaustive search on the indices *i* such that $L_i L'_i \neq 0$ *i.e.* $L_i L'_i = 1$ we will solve the system like a normal HFE system.

3.5 How to make our LL' less expensive in computations

As said above, about r/4 products $L_i L'_i$ are non zero. However sometimes much less products are non zero. In order to be in such a good situation for decryption, one possibility is to send many messages and hope that at least one of them has only a small number of $L_i L'_i \neq 0$. We will see below (sections 5 and 6) how many messages to send in average in order to have a good probability for this to occur.

3.6 Transmission of a session key

Usually a public key encryption algorithm is used not to encrypt a cleartext, but to encrypt a session key that will be used later. Then we can imagine that (with the same public key) we will encrypt potential session keys $k_1, \ldots k_l$ and if k_{α} is successfully decrypted then the receiver, in order to explain that k_{α} was chosen, he will send back a hash of k_{α} (or alternatively, he will use a key derived by k_{α} and send an encrypted text with this key).

Remark. If no send messages has a small number of $L_i L'_i \neq 0$ then the receiver will have to ask for a new set of messages. In order to not leak any information when this occur, it is possible to randomly ask for new messages even if we were able to decrypt the system.

3.7 Transmission of a specific message m

It is mandatory that an opponent do not have much more equations on the clear text than the number of variables of the clear text (otherwise the system becomes much easier to solve with a Gröbner basis attack). But we need to encrypt the message several time in order to have a good probability that for at least one of these ciphertext only a small number of $L_i L'_i$ are non zero.

In order to do so while keeping the same public key we will transform the message M with several codes $M_1, ..., M_k$ with the help of a public transform (such as an AES with public secret keys). Then, the M_i will be encrypted with the public key of our HFE LL'.

3.8 Differences between our LL' and the LL' of [GP06]

in [GP06] a perturbation LL' was also described. However the description differed greatly from ours and it was used in signature rather than encryption. In [GP06] n products $L_i L'_i$ are used rather than a limited number and we simply added them to the public key without any linear combination: $P'_i = P_i + L_i L'_i$. Yet, in this paper we only have r products $L_i L'_i$ and

$$P_j' = P_j + \sum_i \alpha_{i,j} L_i L_i'$$

or

$$A'_j = A_j + \sum_i \alpha_{i,j} L_i L'_i.$$

Moreover in [GP06] not all of the public equations have to be satisfied, unlike in this paper where all the equations are valid equations.

3.9 LL' as a particular case of Hat+

The perturbation Hat+ was introduced in [FmRPP22]. With Hat+ we have r random quadratic equations that are linearly combined on the public (or secret) quadratic equation. In LL' we combine r product $L_iL'_i$. Therefore LL' can be seen as a special case of Hat+.

However our perturbation will be much less costly in terms of computation time than Hat+. This is because $L_i L'_i$ is 0 with a probability 3/4 unlike 1/2 for Hat+.

The article [FmRPP22] gave another description for Hat+: let t be the parameter of the modifier, let $\beta_i \in \mathbb{F}_{q^n}$ for $i \in \{1 \dots t\}$ be random elements, and $\hat{p}_i(x) = \operatorname{Tr}_n\left(\sum_{j,k} \alpha_{i,j,k} x^{q^j+q^k}\right)$ where $\alpha_{i,j,k}$ are random element of \mathbb{F}_{q^n} and let $Q(x) = \sum_i \beta_i \hat{p}_i(x)$. The central map of the modifier is H(x) = F(x) + Q(x) where F(x) is the central map of a vanilla HFE. Here with LL' instead of a random polynomial $\hat{p}_i(x)$ we will take a product of polynomial of the form : $L(x) = \sum_i a_i x^{q^i}$. It shows clearly the link between the two perturbations. And shows that Hat+ is more general than LL'. However we will later show that it does not appear to induce any weaknesses to LL' compared to Hat+. Indeed the cryptanalysis of the two perturbations shows similar results.

In our toy examples we will keep the first representation of the perturbation. But for our cryptanalysis we will prefer the second.

4 Cryptanalysis of LL'

4.1 General considerations

In this section 4 we will study how LL' resist against the classical known attacks in multivariate cryptography (such as MinRank and Gröbner attacks). We may be afraid by the fact that in our schemes, sometimes almost all, or even all, the terms $L_iL'_i$ are 0 on a given cyphertext. Is this a serious problem for the security ? It does not seems so. On MinRank attacks we study the public equations (including the hidden terms $L_iL'_i$) independently from it's values, so the fact that some $L_iL'_i = 0$ does not intervene. On Gröbner attacks we try to solve the public equations with all the terms $L_iL'_i$, and the fact that we want to decrypt even when all these terms are 0 does not really help since these terms are hidden. This was confirmed on many simulations that we did (cf. section 4.5).

Remark. It looks really difficult for an attacker to be able to recover the equations without the $L_i L'_i$ from cleartext/cyphertext pairs. This is because he does not know which message will be decrypted (*i.e.* which messages have almost all $L_i L'_i = 0$) among all the messages send. And also since the messages (the session key) can be publicly transformed with random values, before encryption.

We will now give more details about LL' can resist all known attacks. Since LL' can be seen as a variant of Hat plus we will conduct a similar analysis as done in [FmRPP22] and compare Hat+ and LL'. First of all we will distinguish two types of attacks, Direct attacks (Gröbner Basis) and key recovery attacks (MinRank attacks).

The study of Hat plus as well as our simulations showed that LL' and Hat plus have a very similar behaviour against both types of attacks. So we will consider that both perturbation have the same impact on the degree of regularity of the polynomial. So it follows the following formula:

$$d_{reg} = (d+r)/2 + 2$$

where r is the number of product of linear forms added to the public equations.

For the MinRank attacks analysis we should distinguish two forms of MinRank that we will note MinRank \mathbf{T} , MinRank \mathbf{S} . The goal of the MinRank will be to recover one of the linear transform of the private key, once one of them is found the rest of the key is "easy" to find. Hence MinRank \mathbf{T} will denote the fact that we will try to recover the matrix \mathbf{T} first and MinRank \mathbf{S} the fact that we will try to recover the matrix \mathbf{S} . Let us define first the MinRank problem.

Definition 1. Let $n, m, r, k \in \mathbb{N}$ and let $\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_k$ $n \times m$ matrices over the field \mathbb{F} . The MinRank problem consists to find u_1, u_2, \dots, u_k over \mathbb{F} such that $\operatorname{rank}(\sum_{i=1}^k u_i \mathbf{M}_i) \leq r$.

In order to rewrite the problem of finding one of the two matrices as a Min-Rank. The goal will to use the public key matrices: $\mathbf{P_1} \dots \mathbf{P_n}$ as the matrices $\mathbf{M_1}, \mathbf{M_2}, \dots, \mathbf{M_k}$ (N.B for MinRank **S** we will not directly use the public key matrices) and to use some of the coefficient of the matrices **T** or **S** as the solution vector.

We should also remind here the matrix writing of the HFE map:

Lemma 1. Let $\mathbf{S}, \mathbf{T} \in M_{n \times n}(\mathbb{F}_q)$ then the public key P can be written

$$P = (\mathbf{P}_1, \dots \mathbf{P}_n) = (\mathbf{S}\mathbf{M}_n \mathbf{H}^{*0} \mathbf{M}_n^t \mathbf{S}^t, \dots, \mathbf{S}\mathbf{M}_n \mathbf{H}^{*n} \mathbf{M}_n^t \mathbf{S}^t) \mathbf{M}_n^{-1} \mathbf{T}_n^t$$

where \mathbf{H}^{*i} is the matrix representation of the q^i th power of the secret polynomial h.

4.2 MinRank T

Let's start with MinRank **T**, this method can be found in [BFP11]:

Let q, n, D be standard HFE parameters, $(\mathbf{P}_1, \dots, \mathbf{P}_n)$ the public key and $\mathbf{T}, \mathbf{S}, \mathbf{H}$ the secret key as defined earlier. Then we have

$$(\mathbf{P}_1,\ldots,\mathbf{P}_n) = (\mathbf{S}\mathbf{M}_n\mathbf{H}^{*0}\mathbf{M}_n^t\mathbf{S}^t,\ldots,\mathbf{S}\mathbf{M}_n\mathbf{H}^{*n}\mathbf{M}_n^t\mathbf{S}^t)\mathbf{M}_n^{-1}\mathbf{T}_n$$

So we can write

$$(\mathbf{P}_1,\ldots,\mathbf{P}_n)\mathbf{T}^{-1}\mathbf{M}_n = (\mathbf{S}\mathbf{M}_n\mathbf{H}^{*0}\mathbf{M}_n^t\mathbf{S}^t,\ldots,\mathbf{S}\mathbf{M}_n\mathbf{H}^{*n-1}\mathbf{M}_n^t\mathbf{S}^t).$$

We will write $\mathbf{U} = \mathbf{T}^{-1}\mathbf{M}_n$ and $\mathbf{W} = \mathbf{S}\mathbf{M}_n$. Then we have

$$(\mathbf{P}_1,\ldots,\mathbf{P}_n)\mathbf{U} = (\mathbf{W}\mathbf{H}^{*0}\mathbf{W}^t,\ldots,\mathbf{W}\mathbf{H}^{*n}\mathbf{W}^t).$$

Let $(u_{0,0}, u_{1,0}, \ldots, u_{n-1,0})$ be the first column of **U** then we have

$$\sum_{i=0}^{n-1} u_{0,i} \mathbf{P}_i = \mathbf{W} \mathbf{H} \mathbf{W}^t.$$

Due to the addition of random polynomials the rank of **H** is likely very high. However if we write these polynomials in the field extension then like for Hat plus. There exist β_1, \ldots, β_r coefficient in \mathbb{F}_{q^n} and $L_1, L'_1, \ldots, L_n, L'_n$ linear forms. Such that the central map h(x) verify the equation:

$$h(x) = f(x) + \sum_{i=0}^{r} \beta_i L_i L'_i$$

where f(x) is vanilla HFE central map. If we want to fully keep a univariate representation we can write. $L_i(x) = Tr(\sum_i a_i x^{q^i})$ (where Tr is the trace function of a field extansion). Hence there exist a linear map $\Pi : \mathbb{F}_{q^n} \to \mathbb{F}_{q^n}$ such that $\forall i, \Pi(\beta_i) = 0$. This map has a degree at least r [FmRPP22]. So we can further write:

$$(\mathbf{P}_1,\ldots,\mathbf{P}_n)\mathbf{U}\mathbf{\Pi}=(\mathbf{W}\mathbf{H'}^{*0}\mathbf{W}^t,\ldots,\mathbf{W}\mathbf{H'}^{*n}\mathbf{W}^t).$$

where $\mathbf{H}' = \mathbf{H}\mathbf{\Pi}$ where then the rank of the right member is at most d + r Let $\mathbf{U}' = \mathbf{U}\mathbf{\Pi}$. Hence

$$\operatorname{rank}(\sum_{i=0}^{n-1} u'_{0,i} \mathbf{P}_i) = \log_q(D) + r$$

which is small so finding the first column of **U** reduces to solve a MinRank instance with k = n and $rank = \log_q(D) + r$ on the matrices $\mathbf{P}_1, \ldots, \mathbf{P}_n$. As we can see the increase of the rank is due to the randomness of the β_i (in the univariate representation of the perturbation), the fact that we took special quadratic polynomial in the form $L_i L'_i$ instead of random quadratic p_i (like in HFE Hat+) does not seems to change the complexity of the attack.

4.3 MinRank S

MinRank**S** was first proposed by Ward Beullens and by Tao *et al.* [TPD21].

Retaining the notations **U** and **W** from the previous attack, we have $(\mathbf{P}_1, \dots, \mathbf{P}_{n-1}) = (\mathbf{W}\mathbf{H}^{*0}\mathbf{W}^t, \dots, \mathbf{W}\mathbf{H}^{*n-1}\mathbf{W}^t)\mathbf{U}^{-1}$. Then we obtain

$$(\mathbf{W}^{-1}\mathbf{P}_{1}\mathbf{W}^{-1,t},\ldots,\mathbf{W}^{-1}\mathbf{P}_{n-1}\mathbf{W}^{-1,t}) = (\mathbf{H}^{*0},\ldots,\mathbf{H}^{*n-1})\mathbf{U}^{-1}.$$

This leads us to study the matrix whose line i is the first line of the matrix H^{*i} in other word it's the first line of the representative matrix of the central map after applying the froebenius morphism i times. Let's call this matrix **G**

We can decompose $\mathbf{H} \mathbf{H} = \mathbf{F} + \mathbf{Q}$. With \mathbf{F} being the central map of a vanilla HFE and \mathbf{Q} the perturbation LL' / +hat. The Froebenius operation being linear we can study the effect of the operation on \mathbf{F} and \mathbf{Q} separately. The form of \mathbf{F}

is sparse then the matrix we will obtain will be of the form:

On the other the matrix by \mathbf{Q} will be of no particular form: $\begin{pmatrix} \mathbf{q}_0 \\ \vdots \\ \mathbf{q} \end{pmatrix}$. However,

We can write that \mathbf{q}_i is the first row of the matrix $\sum_{i=0}^t \beta_i^{q^i} \mathbf{Q}_i$. Indeed \mathbf{Q}_i is the representative matrix of the polynomial $\hat{p}_i(x)$ or $Tr(L_i(x)L'_i(x))$ (respectively in the case of Hat+ and LL') whose image is in \mathbb{F}_q . Hence the polynomial is unchanged by the Froebenius morphism.

$$\mathbf{Z} = (\mathbf{U}^{-1})^t \times \mathbf{G}, \ \mathbf{Z} = (\mathbf{U}^{-1})^t \times \left(\begin{pmatrix} \mathbf{A}_1 \\ 0 \\ \mathbf{A}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_n \end{pmatrix} \right)$$

It means that the rank of the right matrix is at most r and thus the rank of

the matrix
$$\begin{pmatrix} \mathbf{A}_1 \\ 0 \\ \mathbf{A}_2 \end{pmatrix} + \begin{pmatrix} \mathbf{q}_0 \\ \vdots \\ \mathbf{q}_n \end{pmatrix}$$
 is at most $d + r$

Furthermore, using the proposition above and the matrix equation of HFE:

Theorem 1. Let $\mathbf{P}_1, \ldots, \mathbf{P}_n$ matrices of the public key and \mathbf{W} the matrix previously defined. If one notes $(w_0^{-1}, w_1^{-1}, \ldots, w_{n-1}^{-1})$ the first row of the matrix \mathbf{W}^{-1} , and $b_i = (w_0^{-1}, w_1^{-1}, \ldots, w_{n-1}^{-1})\mathbf{P}_i$, then the matrix \mathbf{V} whose rows are the b_i has a rank at most d + t.

Proof. From the previous proposition we know that the rank of $\mathbf{V}\mathbf{W}^{-1^t}$ is bounded by d + r, hence the rank of \mathbf{V} is bounded by d + r

For both MinRank once we recover the coefficient of \mathbf{T} or \mathbf{S} we can recover of the totality of the key like in [BFP11], [TPD21].

As mentioned we have essentially reused the cryptanalysis of HFE Hat+ [FmRPP22]. We have successfully simulated this result using a modified version of the code that was published alongside the article [BBC⁺22]. We did not effectively performed the attack, in other word we did not effectively solved the MinRank. But we computed the rank that appeared in the equation previously mentioned. Which confirms the analysis we have performed. So our results showed the augmentation of rank as we have announced.

We performed our tests on a toy example taking: n = 20, d = 7, LL' = 1, 2, 3and using Hat+ with similar parameters n = 20, d = 7, t = 1, 2, 3.

In the end, we computed the rank of the matrix $\mathbf{V}\mathbf{W}^{-1^{t}}$ and the matrix $\mathbf{U}' = \mathbf{U}\mathbf{\Pi}$. and found that their rank coincided with the theory we have developed.

4.4 Resolution of MinRank

The MinRank problem is solved using the support minor method as described in $[BBB^+22]$, $[BBC^+22]$. The complexity of the resolution do not depend on the type of MinRank (**T** or **S**) but only on the target rank, the Field characteristic and the degree of the extension.

$$\mathcal{O}\left((d+r)(n-1)^4 \binom{2(d+r)+1}{d+r}^2\right).$$

This gives us a good idea of the number of LL' that needs to be added. We then have:

r = 12 for 80 bits of security r = 17 for 100 bits of security r = 24 for 128 bits of security r = 40 for 192 bits of security r = 56 for 256 bits of security

(these numbers are obtained with the assumption that d = 0 which is never the case, therefore in the parameters section these numbers will be slightly reduced).

4.5 Direct attacks (Gröbner)

In this section we describe the behaviour of direct attacks using Gröbner basis against HFE LL'. To study the complexity of the attack, the good notion is the degree of regularity. So we will focus on the behaviour of the degree of regularity of HFE LL' central map.

Number of perturbation	$HFE +^{}$	HFE LL'
0	0.86	0.86
1	1.75	1.65
2	2.17	2.13
3	2.11	2.14
4	2.09	2.09

 Table 1. Time to solve the public key of HFE +hat compared with HFE LL' using a Gröbner basis

The work of Joux and Faugère showed the link between the rank and the degree of regularity. Analysis of Hat+ showed that the degree of regularity was increased by the action of Hat+ (cf. [FmRPP22]). Our tests confirmed this result and showed that LL' had a very similar impact. The computation of the Table 1 have been made on a toy example using q = 3, D = 7, n = 10. It shows that both of perturbations have a similar impact against Gröbner. N.B the results are the average for 10 Gröbner inversion so the fact that the time may slightly decrease is simply due to the usage of small parameters. We have a bound for the degree of regularity of HFE. Ding *et. al* [Kle12] showed that the degree of regularity D_{reg} is such that. $D_{reg} \leq \frac{(q-1)d}{2} + q$. It leads to an evaluation of the complexity of the resolution of the gröbner basis of the system.

In order to find a Gröbner basis for a system we are using algorithms from Jean-Charles Faugère (F4, F5 [Fau99][Fau02]).

As we saw LL' have a good impact against Gröbner attacks. However, we could wonder if their exists vulnerabilities that depends in the choice of the ciphertext. For example, let f be the public key of a HFE LL' and let $X = \{x_1, \ldots, x_n\} Y = f(X)$ be such that $\forall i \leq rL_i(X)L'_i(X) = 0$ then we could wonder if the resolution of the equation Y = f(X) with Gröbner basis is easier. We performed tests on a toy example (n=15, r=3, d=7). And found that it was not different from the general case. Meaning that the fact that $\forall i \leq rL_i(X)L'_i(X) = 0$ does not induce any vulnerabilities.

[With random ciphertext	with chosen ciphertext
	17.3	17.03

Table 2. Time to solve a HFE LL' when the ciphertext is randomly chosen, compared with a ciphertext such that Y = f(X) be such that $\forall i \leq rL_i(X)L'_i(X) = 0$

As we can see in 2 in average we take about 17sec to solve both. We each time we compared the same public key, first with a random ciphertext and then with a chosen one. We also found out that from a given public key no major differences were found for example even if we have a time to solve that greatly differs from the average we still find close resolution time for a random ciphertext and a chosen one: (for example 20.3 sec, 20.6 sec). From a theoretical perspective, this

is not a great surprise as the algorithm that find a Gröbner basis do not involve the x_i 's in the first place. The algorithm of Buchberger involve the reduction of S-polynomials, even if it depends on the choice of Y it does not depend on the choice of X. Then the degree of regularity of such system should not change hence the complexity of the resolution of the system via Gröbner Basis will not change.

4.6 Other attacks

In order to break HFE other attacks exists but none of them can achieve the same efficiency as MinRank or direct attacks. We can however mention differential attacks [FGS05] [DGS07]. These attacks were at the time efficient due to the choice of parameters that was at the time too optimistic. However, due to the Gröbner basis and MinRank attacks for public keys of degree 2, parameters are nowadays chosen to a point that differential attacks are irrelevant for our specific case. In some cases however (like taking a HFE polynomial of degree 3 or higher) differential attacks are still important.

4.7 Why the L_i and $L_i L'_i$ must be kept secret

If the L_i and L'_i linear form are made public then some efficient attacks are possible. We present here two attacks.

- With Gröbner: Sometimes all the products $L_i L'_i$ are 0 on a message send. Then when the 2r values L_i are public, an attacker can make an exhaustive search on r forms L_i or L'_i among these 2r forms, say L_1, \ldots, L_r , and then solve with Gröbner algorithm the system of public equation plus $L_1 = 0, \ldots L_r = 0$. Here, the Gröbner algorithm will be efficient due to these extra equations and to the fact that all the perturbations $L_i L'_i$ vanished. *i.e.* from a ciphertext Y the attacker will found the corresponding cleartext X. Moreover, this attack still works even if wrong linear forms L_i " are given in addiction to the right L_i and L'_i . This attack can also easily be extended when a small number of $L_i L'_i$ are non zero.
- With MinRank: Once we know some L_i we can determine their kernel K_i and their image Im_i . So we can find a projection Π_K such that $L_i \circ \Pi_K = 0$ or a projection Π_{Im} such that $\Pi_{Im} \circ L_i = 0$. Should we compose the projection Π_K it means we can perform an attack on the matrix **S** and find a rank of *d* instead of d + r. Similarly, we can compose the projection Π_{Im} it means we can perform an attack on the matrix **T** and find a rank of *d* instead of d + r.

5 Number of encryption in function of the number of copies sent

In this section we will answer the question of the average number of message required to be able to decrypt the message depending of the number of $L_i L'_i = 1$ allowed.

In the following we will keep the parameters we obtained in section 4.4 to avoid MinRank attacks i.e.:

- r = 12 for 80 bits of security
- r = 17 for 100 bits of security
- r = 24 for 128 bits of security
- r = 40 for 192 bits of security
- r = 56 for 256 bits of security

5.1 With all $L_i L'_i = 0$

The probability to have all $L_i L'_i = 0$ on a given encryption is $(3/4)^r$. Therefore, we will send about $(4/3)^r$ messages if we want this to occur with a good probability. For example, for 128 bits of expected security we have r=24 this gives approximately 996 messages to be send. In this case the number of decryption will be equal to the number of message send, so 996 decryption (cf. Table 5)

5.2 With only one $L_i L'_i = 1$

The probability to have exactly one $L_i L'_i = 1$ is $(3/4)^{r-1} * (r/4)$. Therefore it is about $(3/4)^r + (3/4)^{r-1} * (r/4)$ for at most one $L_i L'_i = 1$. For 128 bits of security (r=24) this gives a probability of about 0.9% and we will send about 110 messages. In this case the number of decryptions will be about $110 + 110 \times 24 =$ 2750 (cf. Table 5).

5.3 General formula

The probability to have exactly α equalities $L_i L'_i = 1$ is given by the binomial distribution:

$$\left(\frac{3}{4}\right)^{r-\alpha} \left(\frac{1}{4}\right)^{\alpha} \binom{r}{\alpha}.$$

This leads to the Tables 3,4,5,6,7 below.

5.4 Tables of complexities for decryption with quadratic public equations

Table 3 shows orders of magnitude for the number of messages required to be sent and the number of decryption required in function of the number of accepted $L_i L'_i = 1$.

For example with Hat+ we would have required 2^{12} tries but in average we require only half of them to find the right perturbation.

In our example if we send 31 cypher text we have 63% of chance that one of the messages have all of it's $L_iL'_i$ to be null. And if this happens we will find it in approximately 15-16 tries.

For 80 bits of security, the advantage of LL' against Hat+ is that we decrypt 130 times faster while retaining the same security. However, one must take into

Compu	Computations and Transmissions for 80 bits of expected security					
Number of $L_i L'_i = 1$ Probability for 1 message sent number of messages sent number of dec						
0	3.16%	31.5	31.5			
≤ 1	15.8 %	6.3	81.9			
≤ 2	39.0~%	2.5	198			
≤ 3	64.8 %	1.5	450			
≤ 4	84.1 %	1.2	820			
≤ 5	94.4 %	1.06	1680			

Table 3. Maximal numbers of $L_i L'_i = 1$, number of required messages, number of decryption needed, security 80 bits: r = 12

account the cost of the encryption/decryption using a secret key algorithm and the fact that one must send more cypher text in case of a failure of the decryption. While the advantage is limited it remains significant. The advantage will be much more significant when we increase the number of security bits.

Comput	Computations and Transmissions for 100 bits of expected security					
Number of $L_i L'_i = 1$	Probability for 1 message sent number of messages sent number of decryptio					
0	0.75%	133	133			
≤ 1	5.76~%	17.3	311			
≤ 2	17.12~%	5.8	794			
≤ 3	36~%	2.77	1886			
≤ 4	58.1 %	1.72	4093			
≤ 5	77.3~%	1.29	8007			

Table 4. Maximal numbers of $L_i L'_i = 1$, number of required messages, number of decryption needed, security 100 bits: r = 17

In Table 4 the number of decryption for Hat+ would be $2^{17} = 131072$. So if we send only 17 cypher text we will be 420 times (131072/311 = 420) faster than Hat+.

In Table 5, if we send about 25 cypher text the computing time will be 2200 $(2^{27}/7525)$ times faster than Hat+. The advantages are then significant in regard to Hat+. One must do 7500 decryption which remains realistic. Then 128 bits of security seems reasonable according to the current complexity of the MinRank and Gröbner attacks.

In Table 6 the computation will be about 4 millions times faster than Hat+. Still it requires about 285 000 decryptions. This begins to be quite excessive to be realistically used.

In Table 7, the gains compared to Hat+ are astonishing (8 billions times faster in the case of all $L_iL'_i = 0$) but it is not realistically feasible to send 10 millions messages, or to do 8 millions decryptions. And if we send 462 messages, while the gain is still 33 millions times faster it requires to send 2 billions messages

Comput	Computations and Transmissions for 128 bits of expected security						
Number of $L_i L'_i = 1$	Probability for 1 message sent	number of messages sent	number of decryption				
0	0.1%	996	996				
≤ 1	0.9~%	110	2750				
≤ 2	3.9~%	25	7525				
≤ 3	11.5 %	8.7	20227				
≤ 4	24.6~%	4	51804				
≤ 5	42.15 %	2.3	128000				
≤ 6	60.6~%	1.6	304000				

Table 5. Maximal numbers of $L_iL'_i = 1$, number of required messages, number of decryption needed, security 128 bits: r = 24

Computations and Transmissions for 192 bits of expected security						
Number of $L_i L'_i = 1$	Probability for 1 message sent	number of messages sent	number of decryption			
0	0.001%	100000	100000			
≤ 1	0.014~%	7000	285 000			
≤ 2	0.101 %	984	800 000			
≤ 3	0.47~%	213	$2^{21.2}$			
≤ 4	1.6 %	62	$2^{22.7}$			
≤ 5	4.3 %	23	$2^{24.1}$			
≤ 6	9.6 %	10.4	$2^{25.5}$			
≤ 7	18.1 %	5.5	$2^{26.9}$			
≤ 8	29.9 %	3.3	$2^{28.4}$			

Table 6. Maximal numbers of $L_iL'_i = 1$, number of required messages, number of decryption needed, security 192 bits: r = 40

Comput	Computations and Transmissions for 256 bits of expected security						
Number of $L_i L'_i = 1$	Probability for 1 message sent	number of messages sent	number of decryption				
0	0.00001%	107	$2^{2}3$				
≤ 1	0.0002 %	$5 * 10^5$	$2^{24.8}$				
≤ 2	0.0019~%	52000	$2^{26.4}$				
≤ 3	0.012 %	8200	2^{28}				
≤ 4	0.0058 %	1700	$2^{29.4}$				
≤ 5	0.216 %	462	$2^{31.0}$				
≤ 6	0.665~%	150	$2^{32.4}$				
≤ 7	1.73 %	57	$2^{34.0}$				
≤ 8	3.91 %	25.5	$2^{35.1}$				

Table 7. Maximal numbers of $L_iL'_i = 1$, number of required messages, number of decryption needed, security 256 bits: r = 56

which is too much. Then the 256 bits of security will probably not be attained with the LL' method.

6 Variant with equation of degree 3

If the public equation are of degree 3 (instead of 2) then we can add some perturbations $L_i L'_i L''_i$ (instead of $L_i L'_i$) where L_i , L'_i and L''_i are secret linear forms in x_1, \ldots, x_n . Then the public key will be bigger. For example with $n \approx 200$ the size of the public key in degree 3 will be about $\frac{n^4}{6}$ bits, *i.e.* about 32 Megabytes (Compared to $\frac{n^3}{2}$, about 500 kilobytes in degree 2). However, $L_i L'_i L''_i = 0$ with a probability $\frac{7}{8}$ (instead of $\frac{3}{4}$ for $L_i L'_i = 0$ in degree 2). Therefore we can expect a more efficient scheme in terms of computations and number of bits to send.

6.1 Differential attack and number of terms $L_i L'_i L''_i$

Very often, when we study a multivariate scheme of degree 3, after a differential we obtain a similar scheme of degree 2. Hence, very often the study of multivariate degree 3 scheme does not fundamentally change from the one in degree 2.

Let see the results we obtain after a differential. Let $X = (x_1, \ldots, x_n)$ and $A = (a_1, \ldots, a_n)$. Let $f(x) = x_1 x_2 x_3$ (it's a simplified notation of $L_i L'_i L''_i$). Then the differential f(X + A) + f(A) is:

$$(x_1 + a_1)(x_2 + a_2)(x_3 + a_3) + x_1x_2x_3$$

 $= a_1 x_2 x_3 + a_2 x_1 x_3 + a_3 x_1 x_2$

(we ignore terms of degree 1). If $a_1 = a_2 = a_3 = 0$ then the rank of this expression is 0. If only one of the a_i is non zero, then the rank is 1. If two are non zero, then the rank is also 1, since $x_2x_3 + x_1x_3 = x_3(x_2 + x_1)$. If $a_1 = a_2 = a_3 = 1$ then the rank is 2, since $x_2x_3 + x_1x_3 + x_2x_1 = x_2x_1 + x_3(x_2 + x_1)$. Therefore, in average, the rank will be:

$$0 \times \frac{1}{8} + 1 \times \frac{6}{8} + 2 \times \frac{1}{8} = 1.$$

This means that if we use r terms $L_i L'_i L''_i$, then in the differential in average we will obtain r terms $M_i M'_i$ with M_i and M'_i of degree one. Therefore, we should keep in degree 3 about the same value r as before in degree 2 for the same level of security.

For example for 128 bits of security we will keep $r \approx 24$, as we can see in the Table 8. This Table 8 can be compared with Table 5 (in degree 2). For example in Table 8 we can send 25 messages and only 25 decryptions will be needed (instead of 7525 decryptions on 25 messages in Table 5). The only drawback would be the size of the public key of about 32 Megabyte, if $n \approx 200$ instead of

Computations and Transmissions for 128 bits of expected security in degree 3					
Number of $L_i L'_i L''_i = 1$	Probability for 1 message sent	number of messages sent	number of decryption		
0	4.05%	24.6	24.6		
≤ 1	17.9 %	5.56	139		
≤ 2	40.1 %	2.45	737		
≤ 3	64~%	1.56	3627		

Table 8. Maximal numbers of $L_iL'_i = 1$, number of required messages, number of decryption needed, security 128 bits: r = 24 (degree 3)

Computations	Computations and Transmissions for 256 bits of expected security in degree 3					
Number of $L_i L'_i = 1$	Probability for 1 message sent	number of messages sent	number of decryption			
0	0.056%	1768	1768			
≤ 1	0.509~%	196	11172			
≤ 2	0.0019~%	44	70268			

Table 9. Maximal numbers of $L_i L'_i = 1$, number of required messages, number of decryption needed, security 256 bits: r = 56 (degree 3)

about 500KB in degree 2. Moreover, the study of systems of degree 3 is rather poor compared to the prolific study of degree 2 system. Therefore, there may be better cryptanalysis possible.

From Table 9 we see that for 256 bits of expected security, even with public equation of degree 3, the complexities are really high (probably not realistic, except in for very special usage): Public key of about 32 Megabyte ($n \approx 200$), about 1768 messages to send and about 1768 decryptions.

With public equation of degree 4 much less computations are required but the public key becomes unrealistic $(\frac{n^5}{24} \approx 1.55 \text{ GB} \text{ for } n \approx 200)$. And then again, better cryptanalysis may exist in degree 4.

7 Examples of parameters and performances with public equations of degree 2

In this section we will present some explicit examples for our scheme HFE LL' with parameters and it's performance.

We will present parameters for five level of expected security: 80, 100, 128, 192, 256 bits of security. We realised our tests of performances using the code for GeMSS (more precisely the reference implementation as it allows for flexible parameters choices). Although the code was designed for signatures, it gives us a reasonable indication for the time to decrypt (as the process of decryption is similar to the one to sign). The code does not directly model the perturbation LL' so we multiply the number of cycles we obtained by the average number of decryption required for each level of security. We chose the parameters in order to be protected from the attacks we previously presented.

For clarity we have chosen here only one parameter for the maximal number of $L_i L'_i = 1$ but as shown in the tables of section 5 other trade-off are possible. Here, we have chosen a HFE central map with public key of degree 2 and a secret polynomial of degree 17. With such central map the rank, without perturbation, is 5 (and not 0), it explains why in Tables 10 and 11 we used a parameter r slightly smaller than in section 5.

So we have to obtain parameters such that :

$$2^{s} \le \left((d+r)(n-1)^{4} \binom{2(d+r)+1}{d+r}^{2} \right).$$

Where s is the security parameter and such that:

$$2^s \le \binom{n+d_{reg}}{n}^{\omega},$$

with

$$d_{reg} \approx \frac{d+r}{2} + 2.$$

Security bits	(q,n,D,r,e)	Decrypt one message (Cycles)	Avg. nbr. of decryptions	Total cycles	Total time (s)
80	(2, 100, 17, 10, 0)	2M	17.7	35.5M	0.011
100	(2, 110, 17, 14, 0)	2M	56	112M	0.035
128	(2, 138, 17, 22, 1)	2M	1700	3400M	1.06
192	(2, 200, 17, 38, 1)	3M	168883	506649	158.32
256	(2, 266, 17, 52, 2)	4M	28530507	114122030	35663.13

Table 10. Computation time of a HFELL' scheme, q is the characteristic, n the size of the field extension, D the degree of the central map, r the number of $L_i L'_i$, and e the maximal number of $L_i L'_i = 1$. The total time is an estimation.

Security bits	(q,n,D,r,e)	pk (KB)	Cyphertext of one cyphertext(b)	number of messages	Total size
80	(2, 100, 17, 10, 0)	45	100	17.7	1770.1
100	(2, 110, 17, 14, 0)	61	110	56.12	6173.2
128	(2, 138, 17, 22, 1)	163	138	73	10198.2
192	(2, 200, 17, 38, 1)	497	200	4330	866066.0
256	(2, 266, 17, 52, 2)	1171	266	20689	5503274

Table 11. Sizes of the cyphertext send for a HFELL' schemes, q is the characteristic, n the size of the field extension, D the degree of the central map, r the number of $L_i L'_i$, and e the maximal number of $L_i L'_i = 1$.

In Table 10 we can see that up until 128 bits we have reasonable decryption time. Benchmarking was done on an Intel Core i7-10850H 3.2GHz CPU with

level of security	MinRank	Direct attacks
80	84	80.9
100	101	103
128	134	129
192	200	199
256	258	252

Table 12. Complexity of MinRank attack and direct attacks on our parameters (the complexity of Gröbner attack is optimistic)

32GB of RAM. We can note that the computations are easily parallelisable. We could also use a different central map than HFE. For example we could use a C^* ([MI88]) central map, that is known to be faster than HFE. However, as we already mentioned, all attempts to repair C^{*} have failed. Therefore we are much more confident with HFE LL' than with C^{*} LL' where a specific analysis would be required.

As far as we know the cryptanalysis of HFE LL' does not differ much from the results obtained for HFE +hat as we showed earlier. Hence, we are confident with the current parameters according to the current state of attacks as we can see in Table 12 (the complexity of the attack is computed using the parameters as stated in Table 10).Unless a new idea is discovered and a vulnerability exploits the particular form of the polynomial we add, we should not expect any major changes for our parameters.

8 Conclusion

Very few efficient and secure multivariate encryption algorithms exist at present. This is mainly due to the fast progress of MinRank attacks (to recover the secret key) or Gröbner attacks (to inverse the system). New type of attack (MinRank on S cf. [TPD21]) and new method of resolution (support minor [BBB+22]) have threatened the security of HFE and also threatened the security of most multivariate encryption algorithm (many of them are based on variants of HFE). For example, Vanilla HFE is not efficient when we want more than 80 bits of security.

In this paper, we have presented a new perturbation, called LL', to enforce the security of a multivariate encryption scheme, for example HFE. LL' was previously used for signatures, but in a very different way cf.[GP06]. With our perturbation LL' we have been able to design efficient schemes up to 128 bits of security. (For 128 bits of expected security we need about 1s to decrypt with public equations of degree 2, and the encryption is very fast, as usually in multivariate cryptography). It may also be possible to have more security bits (by using public equations of degree 3 or 4), but then the public key become very large.

Due to the recent introduction of this algorithm, we do not recommend using this scheme right now for very sensible application since many multivariate scheme have been broken in the past. However it is interesting to notice that new ideas can still be found to reinforce the security of multivariate scheme. It is also interesting to notice that, again, it seems possible to use multivariate scheme for encryption.

References

- BBB⁺22. Magali Bardet, Pierre Briaud, Maxime Bros, Philippe Gaborit, and Jean-Pierre Tillich. Revisiting algebraic attacks on MinRank and on the rank decoding problem. Cryptology ePrint Archive, Report 2022/1031, 2022. https://eprint.iacr.org/2022/1031.
- BBC⁺22. John Baena, Pierre Briaud, Daniel Cabarcas, Ray A. Perlner, Daniel Smith-Tone, and Javier A. Verbel. Improving support-minors rank attacks: Applications to GeMSS and rainbow. In Yevgeniy Dodis and Thomas Shrimpton, editors, CRYPTO 2022, Part III, volume 13509 of LNCS, pages 376–405. Springer, Heidelberg, August 2022.
- BFP11. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Cryptanalysis of multivariate and odd-characteristic HFE variants. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 441–458. Springer, Heidelberg, March 2011.
 DM00
- BM09. Olivier Billet and Gilles Macario-Rat. Cryptanalysis of the square cryptosystems. In Mitsuru Matsui, editor, ASIACRYPT 2009, volume 5912 of LNCS, pages 451–468. Springer, Heidelberg, December 2009.
- CBD⁺09. Crystal Clough, John Baena, Jintai Ding, Bo-Yin Yang, and Ming-Shing Chen. Square, a new multivariate encryption scheme. In Marc Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 252–264. Springer, Heidelberg, April 2009.
- CCFST24. Max Cartor, Ryann Cartor, Hiroki Furue, and Daniel Smith-Tone. Improved cryptanalysis of hferp. In Qiang Tang and Vanessa Teague, editors, Public-Key Cryptography PKC 2024, pages 413–440, Cham, 2024. Springer Nature Switzerland.
- CMRPV24. Benoît Cogliati, Gilles Macariot-Rat, Jacques Patarin, and Pierre Varjabedian. State of the art of hfe variants. In Markku-Juhani Saarinen and Daniel Smith-Tone, editors, *Post-Quantum Cryptography*, pages 144–167, Cham, 2024. Springer Nature Switzerland.
- DGS07. Vivien Dubois, Louis Granboulan, and Jacques Stern. Cryptanalysis of HFE with internal perturbation. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 249–265. Springer, Heidelberg, April 2007.
- Fau99. Jean-Charles Faugère. A new efficient algorithm for computing gröbner bases (f4). Journal of Pure and Applied Algebra, 139(1):61–88, 1999.
- Fau02. Jean Charles Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ISSAC '02, page 75–83, New York, NY, USA, 2002. Association for Computing Machinery.
- FGS05. Pierre-Alain Fouque, Louis Granboulan, and Jacques Stern. Differential cryptanalysis for multivariate schemes. In Ronald Cramer, editor, *EURO-CRYPT 2005*, volume 3494 of *LNCS*, pages 341–353. Springer, Heidelberg, May 2005.

- FmRPP22. Jean-Charles Faugère, Gilles macario Rat, Jacques Patarin, and Ludovic Perret. A new perturbation for multivariate public key schemes such as HFE and UOV. Cryptology ePrint Archive, Report 2022/203, 2022. https://eprint.iacr.org/2022/203.
- GP06. Aline Gouget and Jacques Patarin. Probabilistic multivariate cryptography. In Phong Q. Nguyen, editor, *Progress in Cryptology - VIETCRYPT* 06, volume 4341 of *LNCS*, pages 1–18. Springer, Heidelberg, September 2006.
- IPS⁺18. Yasuhiko Ikematsu, Ray A. Perlner, Daniel Smith-Tone, Tsuyoshi Takagi, and Jeremy Vates. HFERP A new multivariate encryption scheme. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography* 9th International Conference, PQCrypto 2018, pages 396–416. Springer, Heidelberg, 2018.
- Kle12. Thorsten Kleinjung. Degree of regularity for hfe minus (hfe-) jintai ding and. 2012.
- KPG99. Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *EUROCRYPT'99*, volume 1592 of *LNCS*, pages 206–222. Springer, Heidelberg, May 1999.
- MI88. Tsutomu Matsumoto and Hideki Imai. Public quadratic polynominaltuples for efficient signature-verification and message-encryption. In C. G. Günther, editor, *EUROCRYPT'88*, volume 330 of *LNCS*, pages 419–453. Springer, Heidelberg, May 1988.
- Pat95. Jacques Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of eurocrypt'88. In Don Coppersmith, editor, CRYPTO'95, volume 963 of LNCS, pages 248–261. Springer, Heidelberg, August 1995.
- Pat96. Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 33–48. Springer, Heidelberg, May 1996.
- SDP16. Alan Szepieniec, Jintai Ding, and Bart Preneel. Extension field cancellation: A new central trapdoor for multivariate quadratic systems. In Tsuyoshi Takagi, editor, Post-Quantum Cryptography - 7th International Workshop, PQCrypto 2016, pages 182–196. Springer, Heidelberg, 2016.
- TPD21. Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. Efficient key recovery for all HFE signature variants. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 70–93, Virtual Event, August 2021. Springer, Heidelberg.