Odd-Char Multivariate Hidden Field Equations

Chia-Hsin Owen Chen[1], Ming-Shing Chen[1], Jintai Ding[2], Fabian Werner[3], and Bo-Yin Yang[2]

[1] IIS, Academia Sinica, Taipei, Taiwan, {by,owenhsin,mschen}@crypto.tw
[2] Dept. of Math. Sci., University of Cincinnati, Cincinnati, Ohio, ding@math.uc.edu
[3] Dept. of Comp. Sci., Technische Universität Darmstadt, Germany, jak@cccmz.de

Abstract. We present a multivariate version of Hidden Field Equations (HFE) over a *finite field of odd characteristic*, with an extra "embedding" modifier. Combining these known ideas makes our new MPKC (multivariate public key cryptosystem) more efficient and scalable than any other extant multivariate encryption scheme.
Switching to odd characteristics in HFE-like schemes affects how an attacker can make use of field equations. Extensive empirical tests (using MAGMA-2.14, the best commercially available $\mathbf{F_4}$ implementation) suggests that our new construction is indeed secure against algebraic attacks using Gröbner Basis algorithms. The "embedding" serves both to narrow down choices of pre-images and to guard against a possible Kipnis-Shamir type (rank) attack. We may hence reasonably argue that for practical sizes, prior attacks take exponential time.
We demonstrate that our construction is in fact efficient by implementing practical-sized examples of our "odd-char HFE" with 3 variables ("THFE") over $\mathbb{F}_{31}$. To be precise, our preliminary THFE implementation is 15×–20× the speed of RSA-1024.

Keywords: HFE, Gröbner basis, multivariate public key cryptosystem

1 Introduction

MPKCs (multivariate public key cryptosystems) [15, 34] is often considered a significant possibility for Post-Quantum Cryptography, with potential to resist future attacks with quantum computers. We discuss both theoretical and practical issues in building an improved multivariate encryption scheme related to HFE (Hidden Field Equations — executive summary in Sec. 2).

1.1 Questions

HFE and related cryptosystems is a major family of MPKCs proposed by Jacques Patarin at Eurocrypt'96 [33]. The basic idea is similar to Matsumoto-Imai [30]: A polynomial of a special form in a large field is transformed and hidden as a system of multivariate quadratic polynomials over a much smaller field.

MPKC's security is often said to rely on the difficulty of solving a system of quadratic multivariate equations over a finite field, which is NP-hard [25] and even conjectured to be hard on average, but the public map of an MPKC is by definition not randomly chosen which opens the possibility of effective attacks.

The literature contains many studies on cryptanalysis of HFE and related systems (e.g., [9,10,24,29]). Today cryptanalyzing basic HFE is regarded as subexponential in time complexity [27]. Projected-secure instances of HFE derivatives are usually too large for practical use. Given the glaring lack of well-regarded multivariate encryption schemes, it seems obvious to ask: "Can we create an HFE variant that is both efficient and resistant to known attacks?"

1.2 Our Answers

We suggest going in a different direction, using a multivariate variant of HFE in a finite field of odd characteristic. Indeed, we can build a very efficient MPKC (Secs. 3 and 5) using just a 3-variable random central map $\overline{\mathcal{Q}} : (X_1, X_2, X_3) \mapsto (Y_1, Y_2, Y_3)$, where $Y_\ell = Q_\ell(X_1, X_2, X_3)$, for $\ell = 1, 2, 3$. X_i and Y_j are elements of the intermediate finite field $\mathbb{L} = \mathbb{F}_{q^k}$, with an odd characteristic.

$$Q_\ell(X_1, X_2, X_3) = \sum_{1\le i\le j\le 3} \alpha_{ij}^{(\ell)} X_i X_j + \sum_{j=1}^{3} \beta_j^{(\ell)} X_j + \gamma^{(\ell)}, \text{ for } \ell = 1,\, 2,\, 3,$$

specifies three randomly chosen quadratic polynomials over $\mathbb{L}$ that determines $\overline{\mathcal{Q}}$. We can write X_i and Y_j with k components each in $\mathbb{K} = \mathbb{F}_q$. The structure of $\overline{\mathcal{Q}}$ is hidden by affine maps in $\mathbb{K}^{3k}$ on either side like in any MPKC. Of course, the same basic design should work with any $h > 1$ (cf. Eq. 4, Sec. 3)

In Sec. 4 we explain why this particular construction should be more secure, in particularly that the higher degree of field equations stifle certain attacks, and show empirical results to be in line with our theoretical predictions.

Our scheme is more efficient because the speed of decryption does not depend directly on the same parameters that determines the security under algebraic attacks. The slowest step during decryption is system-solving for the X_i, which here is essentially a pre-programmed Gröbner basis algorithm. It is fast (cf. Sec. 5.2) due to the small number of variables. Speed tests are summarized in Tab. 1. Note: $h = 2$ and 4 instances included for reference only, we recommend $h = 3$.

Key		128 bits			192 bits		256 bits
CPU		$\mathbb{F}_{13}^{12} \times 3$	$\mathbb{F}_{31}^{15} \times 2$	$\mathbb{F}_{31}^{10} \times 3$	$\mathbb{F}_{31}^{15} \times 3$	$\mathbb{F}_{31}^{10} \times 4$	$\mathbb{F}_{31}^{18} \times 3$
P3	encr	1.13	0.90	0.90	2.90	2.28	5.03
	decr	3.19	1.65	2.86	5.42	22.10	10.76
K8	encr	0.92	0.80	0.80	2.54	1.93	4.38
	decr	2.43	1.25	1.85	4.15	13.53	7.51
C2	encr	0.79	0.70	0.70	2.28	1.78	3.92
	decr	1.93	1.02	1.59	3.28	12.24	6.41

Table 1. Timing Projected Multivariate HFE at various blocksizes on P3, C2, and K8 architectures — measurement numbers are in millions of clock cycles for uniformity

For comparison, RSA-1024 takes 1M cycles to encrypt and 42M cycles to decrypt on the NESSIE final test report [31] running the programs submitted by RSA Security tuned for the P3. At the same time, inverting an HFE polynomial with $d = 129$ in $\mathbb{F}_{2^{103}}$ — the one used in QUARTZ [8], the shortest known signature scheme at 2^{80} design security — takes 1400M cycles.

1.3 Previous Work

There is much literature on the cryptanalysis of HFE-related MPKCs. Overall, most authorities seem to consider HFE and associated schemes insecure:

– We may attack directly using a Gröbner basis system-solver. Faugère broke the HFE challenge 1 set by Patarin [24] by actually solving the system of 80 equations in 80 $\mathbb{F}_2$ variables using his new $\mathbf{F_5}$ algorithm [22]. Today one can reproduce this result using Steel's implementation of $\mathbf{F_4}$ in MAGMA [6, 21]. Later on it was argued that with the new Gröbner basis methods (like $\mathbf{F_4}$ or $\mathbf{F_5}$), HFE has sub-exponential time complexity [27].
– We may try to derive an associated MinRank [9] problem (the Kipnis-Shamir approach) and try to solve that by "relinearlization". Other ideas were proposed to solve the associated MinRank instance [10, 26].
– The abovementioned works imply that if one fixes the parameter D in HFE (degree of the polynomial; more precisely it's $r \approx \lg D$) then the secret key can be found in sub-exponential time as the number n of variables increase.

We hope to show that this is not quite the case when the design is changed. As mentioned earlier, the ideas behind our construction are not really new:

– Multiple hidden quadratic systems in multiple variables in a larger field had been mentioned by Patarin but never seriously explored (as far as we know).
– An MPKC with multiple big-field variables is proposed in [18], but is closer to C^* than to HFE with dissimilar computational and security characteristics.
– Fields of odd characteristic had already been proposed for HFE [17] (and for C^* [34]). However, decryption in the newly proposed HFE instances also requires solving a high degree equation in a large field, which is still slow (cf. speed of QUARTZ in previous section).
– *Embedding* was mentioned as early as [34], and used to strengthen a digital signature scheme in [14] (as *projection*), but here we are doing a practical sized encryption scheme. Sec. 4.4 explains the rationale.

As we prepared this paper, a new preprint [3] was drawn to our attention, where the authors study HFE derivatives similar to ours — but over $\mathbb{F}_{2^k}$. They concluded that these seem no more secure than ordinary HFE. This result is consistent with, and serves to complement, our security arguments in Sec. 4. The need for an odd characteristic, not just a larger base field, is highlighted.

1.4 Summary and Future Work

We think we have a great idea — a simple but very efficient multivariate encryption scheme with potential from both theoretical and practical points of view, more on the practical implementations on this and similar schemes is needed.

Further, we think this work adds to the evidence that the use of field of odd characteristics is a new direction that merits more attention from cryptologists. More theoretical and empirical work should be conducted about Gröbner basis methods when the field equations cannot be efficiently used in the direct attack.

2 MPKCs and HFE

We will first review Multivariate Public-Key Cryptosystems (MPKCs), how classical Hidden Field Equations (HFE) schemes work and how they are attacked.

MPKCs have as public key (the coefficients to) a set of polynomials $\mathcal{P} = (p_1, \ldots, p_m)$ in variables $\mathbf{w} = (w_1, \ldots, w_n)$ where all variables and coefficients are in $\mathbb{K} = \mathbb{F}_q$. $\mathcal{P}(0)$ is always taken to be zero, i.e., public polynomials do not have constant terms. Extant MPKCs almost always hide the private map $\mathcal{Q}$ via composition with two affine maps S, T. So, $\mathcal{P} := T \circ \mathcal{Q} \circ S : \mathbb{K}^n \to \mathbb{K}^m$, or

$$\mathcal{P} : \mathbf{w} = (w_1, \ldots, w_n) \overset{S}{\mapsto} \mathbf{x} = \mathrm{M}_S \mathbf{w} + \mathbf{c}_S \overset{\mathcal{Q}}{\mapsto} \mathbf{y} \overset{T}{\mapsto} \mathbf{z} = \mathrm{M}_T \mathbf{y} + \mathbf{c}_T = (z_1, \ldots, z_m) \tag{1}$$

Different MPKCs have different "central maps" $\mathcal{Q}$. The secret key consists of the information in S, T, and $\mathcal{Q}$, i.e., $(\mathrm{M}_S^{-1}, \mathbf{c}_S)$, $(\mathrm{M}_T^{-1}, \mathbf{c}_T)$ and parameters in $\mathcal{Q}$.

2.1 Hidden Field Equations

Shortly after Patarin defeated the original C^* scheme [32], he proposed his improvement, Hidden Field Equations (HFE) [33]. Like its predecessor C^*, HFE is a "big-field" or "two-field" MPKC. Usually in this type of MPKC, the central map is really a map in a large finite field $\mathbb{L} = \mathbb{F}_{q^n}$, isomorphic to a degree-n extension of $\mathbb{K}$, and expressed as a n-dimensional vector space over $\mathbb{K} = \mathbb{F}_q$. To be quite precise, take $\overline{\mathcal{Q}} : \mathbb{L} \to \mathbb{L}$ (which we can invert), and pick a $\mathbb{K}$-linear bijection $\phi : \mathbb{L} \to \mathbb{K}^n$ and a map $\overline{\mathcal{Q}} : \mathbb{L} \to \mathbb{L}$ that we can invert. Then we have the following multivariate polynomial map, presumably quadratic for efficiency:

$$\mathcal{Q} = \phi \circ \overline{\mathcal{Q}} \circ \phi^{-1}. \tag{2}$$

then, one "hides" this map $\mathcal{Q}$ by composing from both sides by two invertible affine linear maps S and T in $\mathbb{K}^n$, as in Eq. 1. Instead of using for $\mathcal{Q}$ the monomial used by C^*, we use the *extended Dembowski-Ostrom polynomial map*:

$$\overline{\mathcal{Q}} : \mathbf{x} \in \mathbb{L} \longmapsto \mathbf{y} = \sum_{0 \le i \le j < r} a_{ij} \mathbf{x}^{q^i + q^j} + \sum_{0 \le i < r} b_i \mathbf{x}^{q^i} + c \in \mathbb{L}, \tag{3}$$

This map is in general *not* one-to-one; some kind of checksum is required to identify the inverse from one of a number of possible candidates. Inverting

$\mathcal{Q}$ is equivalent to solving a univariate equation of high degree in $\mathbb{L}$. Patarin recommended that the choice for HFE should be $q = 2$ and $n = 128$.

We will henceforth identify $\mathbb{L} = \mathbb{F}_{q^n}$ and the vector space $\mathbb{K}^n = (\mathbb{F}_q)^n$, and hence $\overline{\mathcal{Q}}$ and $\mathcal{Q}$, and omit the mention of ϕ, except when needed.

To decrypt (i.e., to invert $\mathcal{P}$), one needs to invert $\mathcal{Q}$, which is a polynomial over in $\mathbb{L}$ of degree at most $D = 2q^{r-1}$. Therefore we need to solve $\mathcal{Q}(\mathbf{x}) = \mathbf{y}$ efficiently. This is typically accomplished using Berlekamp's algorithm.

2.2 Direct (Algebraic) Attack via Gröbner Bases

To encrypt the plaintext $\mathbf{w} = (w_1, \ldots, w_n)$ using HFE, the user obtains the public key $\mathcal{P}$ of the receiver, computes the ciphertext $\mathbf{z} = (z_1, \ldots, z_n) = \mathcal{P}(w_1, \ldots, w_n)$ and sends it over a usually assumed insecure channel. An attacker who can solve

$$\mathcal{P}(\mathbf{w}) = \mathbf{z}, \text{ or equivalently} z_i = p_i(w_1, \ldots, w_n),\ i = 1 \cdots n,$$

will gain the plaintext $\mathbf{w} = (w_1, \ldots, w_n)$ and hence break the cryptosystem.

The Gröbner basis method [4] is the classical method of solving multivariate polynomial equations. Suppose we wish to solve the set of equations

$$f_1(x_1, \ldots, x_n) = \cdots = f_m(x_1, \ldots, x_n) = 0,$$

over some field $\mathbb{K}$, where $m \geq n$. If we can find a set of polynomials of the form

$$(g_1(x_1, \ldots, x_n),\ g_2(x_2, \ldots, x_n),\ g_3(x_3, \ldots, x_n), \ldots, g_n(x_n))$$

such that the set of polynomials g_i and the set of polynomials f_i generate exactly the same ideal in the polynomial ring. We can then first solve $g_n(x_n) = 0$ for x_n, then substitute that into $g_{n-1}(x_{n-1}, x_n) = 0$ to find x_{n-1}, and so on for all x_i.

When the solutions of this set of equations has dimension 0 — which usually means finitely many solutions, including those over extension fields — the Buchberger algorithm will do exactly that. Indeed, the above essentially defines a Gröbner basis. However, the Buchberger algorithm can be very slow. Two recent improvements by Faugère [21, 22], the $\mathbf{F_4}$ and $\mathbf{F_5}$ algorithms, allows the same system to be solved more reliably and usually more quickly.

Faugère and Joux pointed out [24] that in the process of solving the HFE-associated multivariate quadratic system, the degree of the polynomials generated by $\mathbf{F_4}$ or $\mathbf{F_5}$ should not be higher than $\log D$. This makes the algorithm poly-time for a fixed D, since $\log D$ needs to be small due to the considerations for decryption. The interested reader should refer to Faugère's papers for details.

2.3 Kipnis-Shamir (MinRank-Like) Attacks

The attacker proceeds by moving the problem back to the extension field, where all the underlying structure can be seen. This is a very natural approach if we intend to exploit the design structure of HFE in the attack. To put it simply: Compute the differential $\mathcal{P}(\mathbf{w} + \mathbf{c}) - \mathcal{P}(\mathbf{w}) - \mathcal{P}(\mathbf{c})$ and take its j-th component (which is bilinear in $\mathbf{w}$ and $\mathbf{c}$) as $\mathbf{c}^T H_j \mathbf{w}$. H_k represents the quadratic crossterms in the k-th polynomial of the public key. The minimum rank of linear combinations of the H_i should be exactly r. This is the MinRank problem [5] and is in general exponential, but can be easier if r is small.

Kipnis and Shamir later suggested to take an linear combination of the H_i and take all $(r+1) \times (r+1)$ submatrices to have determinant zero. This leads to a huge assortment of equations. To solve this system, they introduce an idea which they call *relinearization*, which can be considered a special case of the XL solver [11]. It has been argued [10] that using a Lazard-Faugère solver on this system of equations is effective and equally effective as the direct attack.

Like most attacks that works on the field structure, this one is sidestepped if we take out a degree of freedom, and this we will do as well in our design.

3 Design of Multivariate HFE with a Embedding

Fix a small integer $h > 1$ and an odd prime q. Let $\mathbb{K} = \mathbb{F}_q$ and take a degree-k extension $\mathbb{L} = \mathbb{F}_{q^k} \cong \mathbb{K}[t]/(f(t))$ over $\mathbb{K}$, where $f \in \mathbb{K}[t]$ is irreducible. We may also identify $\mathbb{L}$ as the vector space $\mathbb{K}^k$ by the standard map ϕ:

$$\phi : a_0 \ + \ a_1 t \ + \cdots + a_{k-1} t^{k-1} \in \mathbb{L} \mapsto (a_0, \, a_1, \ldots, \, a_{k-1}) \in \mathbb{K}^k.$$

We will randomly choose a $\mathbb{L}^h \to \mathbb{L}^h$ quadratic map

$$\overline{\mathcal{Q}}(X_1, ..., X_h) = (Q_1(X_1, ..., X_h), \cdots, Q_h(X_1, ..., X_h))$$

where each Q_ℓ for $\ell = 1, \ldots, \, h$ is a randomly chosen quadratic polynomial:

$$Q_\ell(X_1, \ldots, X_h) = \sum_{1 \le i \le j \le h} \alpha_{ij}^{(\ell)} X_i X_j + \sum_{j=1}^{h} \beta_j^{(\ell)} X_j + \gamma^{(\ell)}. \tag{4}$$

When h is small, this map $\overline{\mathcal{Q}}$ can be easily inverted as seen below. The induced map $(\phi^{-1} \times \cdots \times \phi^{-1})$ will then trivially collate $(x_1, x_2, \ldots, x_k)$ into $X_1 \in \mathbb{L}$, $(x_{k+1}, x_{k+2}, \ldots, x_{2k})$ into X_2, and so on, and hence convert $\mathbf{x} = (x_1, \ldots, x_{hk})$ into $\boldsymbol{X} = (X_1, \ldots, \, X_h)$. Similarly we can define $\boldsymbol{Y} = (Y_1, \ldots, \, Y_h) = \overline{\mathcal{Q}}(\boldsymbol{X})$.

Convert $\boldsymbol{Y}$ back to $\mathbf{y}$ with $(\phi \times \cdots \times \phi)$ and we have our central map $\mathcal{Q}$. Choose randomly two nondegenerate affine maps S' and T in $\mathbb{K}^{hk}$, and π to be an affine embedding map from $\mathbb{K}^{hk-r}$ to $\mathbb{K}^{hk}$. The public key (map) for our new multivariate encryption scheme is as follows (see Fig. 1):

$$\mathcal{P}(\mathbf{w}) = T \circ (\phi \times \cdots \times \phi) \circ \mathcal{Q} \circ (\phi^{-1} \times \cdots \times \phi^{-1}) \circ S' \circ \pi(\mathbf{w}). \tag{5}$$

Obviously $\mathcal{P} : \mathbb{K}^{hk-r} \to \mathbb{K}^{hk}$ is a quadratic map. In correspondence with standard notation for MPKCs, we have $(n, m) = (hk - r, hk)$ and $S = S' \circ \pi$.

4 Security Analysis for the New Design

We will first argue theoretically why the new scheme could resist known attacks, and then support our claims with experimental results.

$$\begin{tikzcd}
& & \boldsymbol{X} \in \mathbb{L}^h \arrow[r, "\overline{\mathcal{Q}}"] & \boldsymbol{Y} \in \mathbb{L}^h \arrow[d, "\phi\times\cdots\times\phi"] & \\
\mathbf{w} \in \mathbb{K}^{hk-r} \arrow[r, "\pi"] \arrow[rr, bend left, "S"] \arrow[rrrr, bend right, "\text{Public Key } \mathcal{P}"'] & \mathbb{K}^{hk} \arrow[r, "S'"] & \mathbf{x} \in \mathbb{K}^{hk} \arrow[u, "(\phi^{-1}\times\cdots\times\phi^{-1})"] \arrow[r, "\mathcal{Q}"] & \mathbf{y} \in \mathbb{K}^{hk} \arrow[r, "T"] & \mathbf{z} \in \mathbb{K}^{hk}
\end{tikzcd}$$

Fig. 1. The Design of the Embeded Multivariate HFE Scheme

4.1 On Gröbner Basis Algorithms and the Operating Degree

When running a Gröbner basis algorithm ($\mathbf{F_4}/\mathbf{F_5}$/XL), the degree D_0 that reached by $\mathbf{F_4}$ or $\mathbf{F_5}$ when computing the Gröbner basis in the cheap order — or by XL as it terminates — is known as the *operating* or *critical degree* for that instance. D_0 is the most important parameter that decides both the space and time complexities of a computation.

For today's Gröbner basis methods the time complexity is polynomial — between quadratic and cubic — in the total number of monomials N; so is the space requirements poly(N) — between linear and quadratic. For fixed q, if D_0 grows roughly proportional to n (the number of variables) then N (cf. Appendix B) is exponential in n, hence both time and space needed will grow exponentially.

Conversely if $D_0/n \to 0$, the computation is sub-exponential — in a cryptological setting most of the time this means that a scheme can be defeated.

Of course, D_0 is difficult to compute without actually running the algorithm. Explicit formulas are obviously even harder to come by. However, as long as D_0 grows roughly linearly with n, Gröbner basis attacks will be exponential time.

For "suitably generic" systems a good approximation to D_0 is available, given by generating functions (again cf. Appendix B). If one fixes the degrees for each equation and randomly choose every coefficient from $\mathbb{K} = \mathbb{F}_q$, the algorithm will usually terminate at one particular D_{reg}, the *"degree of regularity"*. If in solving a number of systems from a collection a Gröbner basis method, say $\mathbf{F_4}$, usually terminates earlier, we can distinguish between our collection and "generic".

We intend to show thus that Gröbner basis methods won't defeat our variant as it did the original HFE, and won't even serve effectively as a distinguisher.

4.2 Why a Bigger Field and not $\mathbb{F}_2$

A system of n generic quadratics in n variables, counting multiplicities and algebraic extensions, should have exactly 2^n solutions. Consequently, the final reduced Gröbner basis in lex order will have a univariate polynomial of degree 2^n. Hence, computing a Gröbner basis should never stop below degree 2^n.

How is it then that all the direct attacks on HFE have much lower critical degrees [24, 27]? When we only want solutions in the field $\mathbb{K} = \mathbb{F}_q$, we may add to the original system $p_i(w_1, ..., w_{hk-r}) = z_i$ the equations $w_i^q = w_i$ which are satisfied by (and only by) the elements of the field $\mathbb{K}$. With an intelligent computer algebra systems (CAS), e.g. MAGMA, one lists these "field equations" with the system of equations to be solved and the CAS should then intelligently tune its method for $\mathbb{F}_q$, especially if $q = 2$ (e.g., no squares or higher powers).

Approximately [1, 36], the degree of $\mathbb{F}_q$-regularity for a "generic" system of polynomials is close to that of a similarly structured "generic" system in a char$= 0$ field ($\mathbb{Q}$, say) plus one generic equation of degree-q for each variable. Not quite equal, since $w_i^q = w_i$ is not exactly generic. In system-solving, in general a lower-degree equation is more useful than one of higher degree [2, 35]. So as q increases, the field equations helps less. Phrased differently, when a Gröbner basis algorithm is still operating at degree $D < q$, an extra degree-q equation does not matter. When the critical degree D_0 is less than q, the solving process would be almost exactly as if we are operating in $\mathbb{Q}$ or $\mathbb{R}$.

For an example, solving 30 $\mathbb{F}_{11}$-variables should never be helped by field equations in $\mathbf{F_4}$, since a single multivariate polynomials in 30 variables of degree 11 would have 5×10^9 $\mathbb{K}$-elements, and the system can't possibly fit in memory.

No Embedding ($r = 0$) The system is $(p_i(w_1, \ldots, w_{hk}) - z_i)_{i=1,\ldots,hk}$. When we do not use the field equations, a Gröbner basis algorithm would compute all solutions over the algebraic closure of $\mathbb{K}$. The variety should be of dimension 0 and consist of 2^{hk} points, and it is in fact so observed in all our experiments – the last polynomial in the Gröbner basis in fact has degree 2^{hk}.

A modern Gröbner basis computation usually starts in a cheap order (like graded-lex) then converts with FGLM. If the solution set is exponential in size, so is the complexity of FGLM, which is polynomial in the codimension of the ideal (which in our case is usually the same as the number of solutions when as here the ideal is 0-dimensional). Formally it depends on the size of an embedded quotient ring of the polynomial ring which can be proved to be in bijection with the solution set [12, 23]. To be *really* accurate, the complexity of FGLM is at least 2^{hk} if there is a chain of non-standard monomials in lex-order of length 2^{hk}. Since the univariate polynomial $g_{hk}(x_{hk})$ has degree 2^{hk}, a chain of non-standard monomials in x_{hk}, namely $x_{hk} <_{lex} x_{hk}^2 <_{lex} \cdots <_{lex} x_{hk}^{2^{hk}-1}$ exists in the basis of the quotient.

So the attacker is stuck if the size of the solution set cannot be cut down:

– One may compute a Gröbner basis in lex-order directly; but the algorithm runs into polynomials with degrees near 2^{hk} ($hk \gtrsim 30$ in practical range).
– One may compute the basis in a cheap order and then "transfer" via FGLM. But this is not applicable either, because the complexity of FGLM is simply too high. There are other transfer algorithms like the Gröbner walk algorithm, but FGLM is the most efficient and most well-studied, and any transfer algorithm will face the exponentially big solution set.

Embedding or Guessing In order to solve such a system, an attacker must first cut down the size of the solution set. One way is to guess at some variables.

- Guessing shrinks the size of the solution set; this idea is well-known in cryptology (as in FXL [11,35]). Mathematically, the variety is intersected with one or more hyperplanes. Some variables become linear functions of others.
- "Embedding" modifiers has the same effect (with no cost to the attacker).
- Guessing or throwing away variables with "embedding" does indeed cause the solution set (in $\mathbb{K}$ *or* extension fields) to shrink significantly. However, *it also destroys algebraic structure that might lead to easier solutions*, and it is well-known that polynomial systems with very few (even 1) solutions can still be hard to solve. Indeed, when one then runs, say, MAGMA:
  1. First one must compute (using $\mathbf{F_4}$) a Gröbner basis in a graded order; tests show that this takes still exponential time ($\Omega(2^{(\alpha+o(1))hk})$) within practical range of parameters, after up to 3 variables are correctly guessed. In fact, our data (Table 2) shows that without guessing variables, the D_0 (critical degree) of the systems produced by our scheme are exactly the same as D_{reg} for random systems (degree of regularity, cf. Appendix B). When guessing variables, $D_{reg} - D_0$ is at most 1. D_0 is seen to grow roughly linearly with hk for various r and q.
  2. Then one runs FGLM, but experiments (though limited) show that the solution set in extension fields, and hence the max degree in MAGMA is *still* of exponential size. Heuristically, if the maximum extension degree in the solution set is μ, then there are $2^{hk}/q^{\mu}$ solutions in extended fields where $w_{hk} = c \in \mathbb{K}$. Of course μ could be large, but our tests do show that the FGLM timing grows like some $\Omega(2^{\beta hk})$ for hk up to about 20.

In summary: guessing an optimal number of variables is often correct (esp. when storage and parallelization are issues) but it does not change the overall picture when the system is *sufficiently indistinguisable from generic.* System-solving is conjectured to be probabilistically difficult. *Our empirical evidence over a broad range of field sizes shows our polynomial systems to be not only hard to solve, but even hard to distinguish from truly random ones using Gröbner basis methods.* Hence it indicates that our polynomial systems are intrinsically difficult to solve and provide strong support for the security of our cryptosystems.

4.3 Why an Odd Prime Field Size

One may then ask: Why is the above an argument also for $\mathbb{K}$ not a bigger field of characteristic 2? Yet for a system of n variables and m equations in $\mathbb{F}_q$, where $q = 2^e$, one can always tranform this set of equations over $\mathbb{F}_q = \mathbb{F}_{2^e}$ into a set of equations over $\mathbb{F}_2$ with $n \times e$ variables and $m \times e$ equations by treating $\mathbb{F}_q$ as a vector space $\mathbb{F}_2^e$. Then field equations of $\mathbb{F}_2$ again can be used. The previous suggestion of XLF [7] (which works although slower than first claimed) is in essence exactly this idea. Hence, we go for an odd characteristic.

We have corroborating evidence in the preprint [3] where the authors study HFE derivatives similar to ours — but over $\mathbb{F}_{2^k}$, and purports to find it no better than ordinary HFE. This is the *security* argument for a medium-sized prime field for $\mathbb{K}$. There is the *implementation* aspect, which we will mention in Sec. 5.

4.4 Why use an Embedding Map π

Against multivariate HFE with odd characteristics, the so-called Kipnis-Shamir attack can still be used [16] even though some doubts exist concerning its effectiveness [28]. If the K-S attack is not effective against our HFE variant, we need not use embedding at all. But if it works, we must defend against it.

One defense is to toss away all terms in the last variable w_n (or more than one variable if needed) — setting the variable to be zero. The effect is to restrict the input variables $\mathbf{w}$ to a subspace of low co-dimension, on which the hidden field structure(s) used by the K-S attack are destroyed (cf. Sec. 2.3).

This had been used with C^* [14] to construct an SFLASH-analogue resistant to differential attacks of Dubois *et al* [19,20] by breaking the symmetry of the big field. It was called "projection" but we term it "embedding" here as a projection should be surjective. The point is that on this subspace, the big-field structure does not exist anymore and attacks using the field structures are prevented.

4.5 Cryptanalytical Experiments and Results

We implemented the new HFE scheme in MAGMA using $k \in \{4, 5, 6, 7\}$ and $h = 3$. We then encrypted random plaintext $(w_1, ..., w_{hk-r})$ resulting in the ciphertext $(z_1, ..., z_{hk})$. Then we tried to break the scheme by computing the Gröbner basis corresponding to the ideal spanned by the system $p_i(x_1, ..., x_{hk-r}) - z_i$ for $i = 1, ..., hk$, using MAGMA-2.14 on a 4-socket dual-core 2.6GHz Opteron (K8) machine. The process is allowed a maximum of 4GB of memory. Inserting the field equations into the set of equations actually slowed things down, exactly as argued above. The system reacts as expected (see also [17]) in all such tests.

We ran a further set of tests — when simply feeding the system to MAGMA, we artificially guessed some of the variables forming the plaintext correctly, by simply substituting w_{hk-j+1} by its concrete value w_{hk-j+1} for $j = 1, 2, 3$ in every $p_i - z_i$ for $i = 1, \dots, hk$. The system would become overdefined with hk equations but only in $hk-j$ variables. With these tests, we dealt with embedding and guess in one fell swoop because they have the same effect on the variety. Apart from the time needed to compute the Gröbner basis, we have tabulated down the critical degree that occured during the process.

The timing data is in Fig. 2 of Appendix A. Fig. 2 shows the running time for $h = 3$ internal maps. In the figure, we take hk as the X-coordinate (actually marked mn) and the running time (in seconds) as the logarithmic scaled Y-coordinate. Let HFE$_j$ denote an instance of the new HFE scheme as described above with j variables correctly guessed, so it will result in a system in hk equations but only in $hk - j$ variables. The left graph shows HFE$_1$, the one in the middle the results to HFE$_2$ and the right one represents the results for HFE$_3$. Some values are missing in the following tables because the process consumed more than 4GB of memory in which case the process was eventually cancelled. The figure clearly supports our conclusion that the time complexity grows exponentially with increasing k.

The critical degree data is summarized in Tab. 2 of Appendix A. The critical degrees for different systems do not depend on the underlying characteristic (i.e. q) but only on the number of variables hk. This supports our conjecture that the multivariate HFE polynomials appear to be random equations for small numbers of guesses. Here "–" means that the process did not finish because of the memory consumption. RND$_i$ means "random system with i variables guessed".

All in all, interpolating the results show that a security level of 2^{80} should be reached for $h \geq 9$ and $h = 3$, so starting from $n = hk = 27$ variables, it should take the computer more than 2^{80} instructions for computing the Gröbner basis.

5 Implementation Details and Testing

We will assume that the object of public-key encryption is only to transmit a key for symmetric ciphers. For the sake of argument, we used block sizes which are tuned to transmit 128-, 192- and 256-bit AES keys. Only $h = 2, 3$ are tested to be correct code. Our *preliminary* implementations (C++, `gcc -O3`, no architectural primitives) are tested on what's sitting in lab: the AMD Opteron (K8), and the Intel Pentium III (P3) and Core 2 (C2) architectures. Some results in Tab. 1.

All in all, we recommend here $h = 3$ and $r = 1$ (which we call THFE, T is for "Trinity") with $q = 31$ as given in Tab. 1. Other values of h are similar.

Note: *To avoid unnecessarily modulo-q operations and fully exploit the power of modern CPUs, one should use* `short int` *or* `int` *data types. Simple irreducible polynomials, e.g.,* $t^{15} - 3 \bmod 31$, *also helps. The next logical step would be hand optimizing with in-lined assembly or vector (SSE) C instrinsics.*

5.1 Key Generation

Key generation is by the standard process of interpolation [30]. We randomly select M_S, $\mathbf{c}_S$, M_T plus the parameters in $\mathcal{Q}$, and set $\mathbf{c}_T := \mathrm{M}_T \mathcal{Q}(\mathbf{c}_S)$, which makes all the constant terms zero. Now we can evaluate $\mathcal{P}(\mathbf{w}) = T \circ \mathcal{Q} \circ S(\mathbf{w})$ for any $\mathbf{w}$. Write the public key like Matsumoto-Imai:

$$z_\ell = \sum_i w_i \left[P_{i\ell} + Q_{i\ell} w_i + \sum_{j<i} R_{ij\ell} w_j \right]. \tag{6}$$

Let $\mathbf{b}_i \in \mathbb{F}_q^n$ be the unit vector in the i-th axis, and do

$$\begin{aligned} Q_{i\ell} &:= \left(p_\ell(2\mathbf{b}_i) - 2p_\ell(\mathbf{b}_i)\right)/2 \\ P_{i\ell} &:= p_\ell(\mathbf{b}_i) - Q_{i\ell} \\ R_{ij\ell} &:= p_\ell(\mathbf{b}_i + \mathbf{b}_j) - Q_{i\ell} - Q_{j\ell} - P_{i\ell} - P_{j\ell} \end{aligned} \tag{7}$$

So key generation means invoke n^2 times the combination $T \circ \mathcal{Q} \circ S$. We can see that both S and T takes about n^2 time. If $h = 3$, each evaluation of $\mathcal{Q}$ involves 27 multiplications in $\mathbb{F}_{q^k}$, where $k \sim n/3$ and which if we do "schoolbook" multiplication will be about $\propto n^2$ multiplications in $\mathbb{F}_q$. So we see that worst case key generations takes $O(n^4)$ time measured in multiplications in $\mathbb{K} = \mathbb{F}_q$.

5.2 Decryption

Decryption comprise the inversion of each of the stages in the public map. The tricky one is of course the invertion of $\mathcal{Q}$, which is not too hard if h is small. We recommend $h = 2$, 3, or 4. Most of the discussion will default to $h = 3$, even though analogous arguments for $h = 2$ or 4 can be inferred similarly.

Suppose we want to decrypt a ciphertext $\mathbf{z} = (z_1, \ldots, z_m)$, where $m = hk$. We apply T^{-1}, collate $\mathbb{K}$-variables y_i into $\mathbb{L}$-values C_j and get a system to solve:

$$Q_\ell(X_1, \ldots, X_h) = \sum_{1 \le i \le j \le h} \alpha_{ij}^{(\ell)} X_i X_j + \sum_{j=1}^{h} \beta_j^{(\ell)} X_j + \gamma^{(\ell)} = C_\ell,\ \ell = 1 \cdots h. \quad (8)$$

A good idea is to implement a miniature Gröbner basis method, specialized for the occasion. We do a tailored matrix-$\mathbf{F_4}$ here. Working with 3 quadratic equations in 3 variables, we will in succession run a Gaussian elimination on matrices of dimensions 3×10, 11×19, 8×16, 5×13, with many coefficients that we know to be zero in advance. The other obvious choice is to eliminate variables via resultants (Appendix C). Either method reduces to a univariate equation in $\mathbb{L}$, and the next section tells us how to find the roots of this polynomial. Back-substitution will then find full solutions to the system of Eq. 8.

5.3 Summary of Univariate Equation-Solving

Using an odd-prime base field let us implement Cantor-Zassenhaus — *much easier and faster than Berlekemp, but inapplicable for fields $\mathbb{F}_{2^{prime}}$ (i.e.,* QUARTZ) — for finding all solutions in $\mathbb{L} = \mathbb{F}_{q^k}$ to a univariate degree-d equation $u(X) = 0$:

1. Replace $u(X)$ by $\gcd(u(X), X^{q^k} - X)$ so that u splits (factors completely) in $\mathbb{L}$. Most of the work is to compute $X^{q^k} \bmod u(X)$, which can be done by
   (a) Compute serially and tabulate $X^d \bmod u(X), \ldots, X^{2d-2} \bmod u(X)$.
   (b) Compute $X^q \bmod u(X)$ using the square-and-multiply method.
   (c) Compute $X^{qi} \bmod u(X)$ for $i = 2, 3, \ldots, d-1$ using the above result.
   (d) Compute $X^{q^i} \bmod u(X)$ for $i = 2, 3, \ldots, k$ with the help of the map $X \mapsto X^q$ in $\mathbb{L}$, this is a precomputable linear map over $\mathbb{K} = \mathbb{F}_q$.

   If using schoolbook multiplication, this takes cubic time in $\mathbb{L}$-multiplications in $(d, k, \lg q)$; (a–d) uses respectively d^2, $3d^2(\lg q)$, $2d^3$, and $2k\, d^2$ mults.
2. Take a random polynomial $v(X)$ such that $\deg v = \deg u - 1$, compute $\gcd\left(v(X)^{(q^k-1)/2} - 1, u(X)\right)$, repeat until u is nontrivially factored.

   This takes also $(3 \lg q + 4k)\, d^2$ $\mathbb{L}$-multiplications, for a total of work quintic in $(d, k, \lg q)$ overall, which affects our choices of parameters a little.

Given the usual selection of parameters, the most time-consuming parts are 1(d) and 2, both of which are roughly $O(kd^2)$ $\mathbb{L}$-multiplications or $O(k^3 d^2)$ $\mathbb{K}$-mults.

5.4 Parameter Choices, Setup and Encryption

We choose $q = 31$ as a suitable compromise between not too small q (degree of the Gröbner basis method argument, and the need to compute on too many $\mathbb{F}_q$-components) and not too large q (makes computation difficult). Note that setting up can be cumbersome — converting between packed and unpacked forms of keys (public *plus* private) can take some 500,000 cycles.

To encrypt, take an AES key of the appropriate length and convert to base $q = 31$, then pad if necessary. For example, suppose we are using $h = 3$ variables in $\mathbb{F}_{31^9}$ to transmit a 128-bit key, then since $\log_2 31^{27} \gtrsim 133.76$, if we convert a 128-bit number to a 27-digit base-31 number, one digit will always be zero. This is fine since we are doing embedding at the same time. For purely cosmetic reasons, we choose always the last variable(s) that that is thrown away (in this case w_{26}). When there is room for some extra bits of randomness after embedding, we always fill with chunks of SHA-1 values of the input.

As shown in Tab. 1, we can see that the conservative instance with $h = 3$ and $\mathbb{L} = \mathbb{F}_{31^{10}}$ to be about 15 times as fast as RSA-1024 on the same computer. With $h = 2$ and $\mathbb{L} = \mathbb{F}_{31^{15}}$ it is a further 1.5× faster. Using $h = 4$ and $\mathbb{L} = \mathbb{F}_{31^{10}}$ is a lot slower, but still about twice as fast as RSA-1024.

This seems a good time to mention some design choices. One obvious question is “why not a smaller q, say $q = 13$” (also tested). The answer is twofold. First, it now requires more blocks to transmit the same sized key, which on balance makes it appreciably slower. Secondly, we also don't want the operating degree D_0 to exceed q too quickly because we wish to play safe. Reasons of security is also why we recommend $h = 3$. Using $h = 2$ makes each multiplication in $\mathbb{L}$ take more than twice the time. Since tests with $h = 2$ only show a 50% speed gain, and we have neither run our Gröbner basis tests extensively with values of h other than 3, nor understand all the theoretical complexities, we stay with $h = 3$ — seemingly a good compromise — and leave $h = 2$ and 4 for future work.

References

1. M. Bardet, J.-C. Faugère, and B. Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *Proceedings of the International Conference on Polynomial System Solving*, pages 71–74, 2004. Previously INRIA report RR-5049.
2. M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic expansion of the degree of regularity for semi-regular systems of equations. In P. Gianni, editor, *MEGA 2005 Sardinia (Italy)*, 2005.
3. O. Billet, J. Patarin, and Y. Seurin. Analysis of intermediate field systems. presented at SCC 2008, Beijing.
4. B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Innsbruck, 1965.
5. J. F. Buss, G. S. Frandsen, and J. O. Shallit. The computational complexity of some problems of linear algebra. Research Series RS-96-33, BRICS, Department of Computer Science, University of Aarhus, Sept. 1996. `http://www.brics.dk/RS/96/33/`, 39 pages.

6. Computational Algebra Group, University of Sydney. The MAGMA computational algebra system for algebra, number theory and geometry. http://magma.maths.usyd.edu.au/magma/, 2005.
7. N. Courtois. Algebraic attacks over $GF(2^{\mathrm{k}})$, application to HFE challenge 2 and SFLASH-v2. In *Public Key Cryptography — PKC 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 201–217. Feng Bao, Robert H. Deng, and Jianying Zhou (editors), Springer, 2004. ISBN 3-540-21018-0.
8. N. Courtois, L. Goubin, and J. Patarin. *Quartz: Primitive specification (second revised version)*, Oct. 2001. `https://www.cosic.esat.kuleuven.be/nessie` Submissions, Quartz, 18 pages.
9. N. T. Courtois. Efficient zero-knowledge authentication based on a linear algebra problem minrank. In *Advances in Cryptology — ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 402–421. Colin Boyd, ed., Springer, 2001.
10. N. T. Courtois, M. Daum, and P. 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, ed., Springer, 2002. `http://eprint.iacr.org/2002/138`.
11. N. T. Courtois, A. Klimov, J. Patarin, and A. 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, ed., Springer, 2000. Extended Version: `http://www.minrank.org/xlfull.pdf`.
12. D. A. Cox, J. Little, and D. O'Shea. *Using algebraic geometry*, volume 185 of *Graduate Texts in Mathematics*. Springer, 2005. 2nd ed.
13. C. Diem. The XL-algorithm and a conjecture from commutative algebra. In *Advances in Cryptology — ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 323–337. Pil Joong Lee, ed., Springer, 2004. ISBN 3-540-23975-8.
14. J. Ding, V. Dubois, B.-Y. Yang, C.-H. O. Chen, and C.-M. Cheng. Could SFLASH be repaired? In L. Aceto, I. Damgard, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *ICALP (2)*, volume 5126 of *Lecture Notes in Computer Science*, pages 691–701. Springer, 2008. E-Print 2007/366.
15. J. Ding, J. Gower, and D. Schmidt. *Multivariate Public-Key Cryptosystems*. Advances in Information Security. Springer, 2006. ISBN 0-387-32229-9.
16. J. Ding and D. Schmidt. Cryptanalysis of HFEv and internal perturbation of HFE. In *Public Key Cryptography — PKC 2005*, volume 3386 of *Lecture Notes in Computer Science*, pages 288–301. Serge Vaudenay, ed., Springer, 2005.
17. J. Ding, D. Schmidt, and F. Werner. Algebraic attack on hfe revisited. In *ISC 2008*, Lecture Notes in Computer Science. Springer. to appear.
18. J. Ding, C. Wolf, and B.-Y. Yang. ℓ-invertible cycles for multivariate quadratic public key cryptography. In *PKC*, volume 4450 of *LNCS*, pages 266–281. Springer, April 2007.
19. V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. Practical cryptanalysis of SFLASH. In *Advances in Cryptology — CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 1–12. Alfred Menezes, ed., Springer, 2007.
20. V. Dubois, P.-A. Fouque, and J. Stern. Cryptanalysis of SFLASH with slightly modified parameters. In M. Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 264–275. Springer, 2007.

21. J.-C. Faugère. A new efficient algorithm for computing Gröbner bases (F_4). *Journal of Pure and Applied Algebra*, 139:61–88, June 1999.
22. J.-C. Faugère. A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In *International Symposium on Symbolic and Algebraic Computation — ISSAC 2002*, pages 75–83. ACM Press, July 2002.
23. J.-C. Faugère, P. M. Gianni, D. Lazard, and T. Mora. Efficient computation of zero-dimensional Gröbner bases by change of ordering. *Journal of Symbolic Computation*, 16(4):329–344, 1993.
24. J.-C. Faugère and A. 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, ed., Springer, 2003.
25. M. R. Garey and D. 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.
26. L. Goubin and N. T. Courtois. Cryptanalysis of the TTM cryptosystem. In *Advances in Cryptology — ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 44–57. Tatsuaki Okamoto, ed., Springer, 2000.
27. L. Granboulan, A. Joux, and J. Stern. Inverting HFE is quasipolynomial. In C. Dwork, editor, *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 345–356. Springer, 2006.
28. X. Jiang, J. Ding, and L. Hu. Kipnis-shamir attack on hfe revisited. In *Inscrypt 2007*, volume 4990 of *Lecture Notes in Computer Science*, pages 399–411. Springer, 2008. Cryptology ePrint Archive, Report 2007/203.
29. A. Kipnis and A. 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, ed., Springer, 1999. `http://www.minrank.org/hfesubreg.ps` or `http://citeseer.nj.nec.com/kipnis99cryptanalysis.html`.
30. T. Matsumoto and H. 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, ed., Springer, 1988.
31. NESSIE: New European Schemes for Signatures, Integrity, and Encryption. Information Society Technologies programme of the European commission (IST-1999-12324). `http://www.cryptonessie.org/`.
32. J. 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, ed., Springer, 1995.
33. J. 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, ed., Springer, 1996. Extended Version: `http://www.minrank.org/hfe.pdf`.
34. C. Wolf and B. 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.
35. B.-Y. Yang and J.-M. Chen. All in the XL family: Theory and practice. In *ICISC 2004*, volume 3506 of *Lecture Notes in Computer Science*, pages 67–86. Springer, 2004.

36. B.-Y. Yang and J.-M. Chen. Theoretical analysis of XL over small fields. In *ACISP 2004*, volume 3108 of *Lecture Notes in Computer Science*, pages 277–288. Springer, 2004.

A Empirical Data: Timings and Critical Degrees

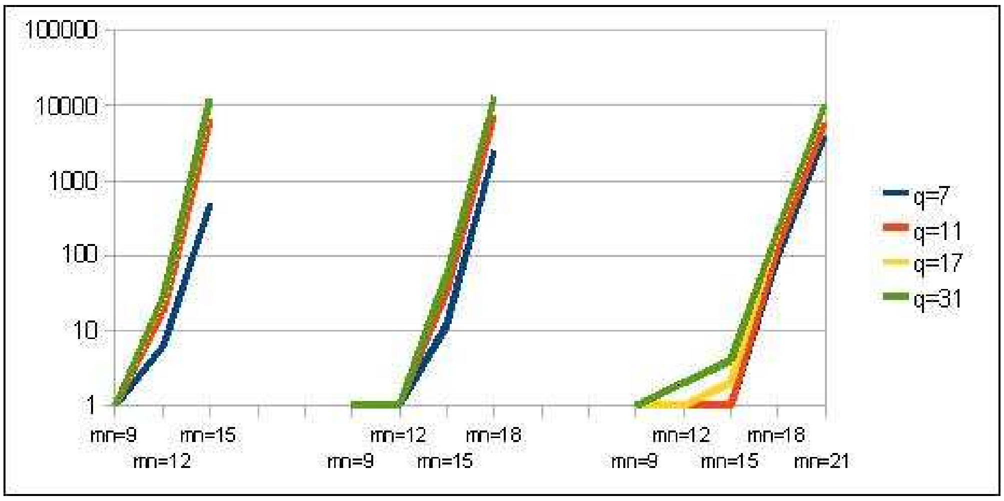

Fig. 2. Timings to break multivariate HFE systems when $q = 7, 11, 17, 31$

Fig. 2 is the time data for our experiments. Table 2 is the data of the critical degree. Here the experiments are for $q = 11, 17, 31$, which have the same results. THFE$_i$ stands for the data where i is the number of variables of values guessed. RNS$_3$ stands for a corresponding random system with 3 variables guessed. This data shows that our system bahaves essentially the same as the random system up to a high degree and computationally it is not possible to distinguish them.

Scheme\ $hk =$	9	12	15	18
THFE$_1$	9	12	14	–
THFE$_2$	8	12	13	–
THFE$_3$	8	11	13	16
RND$_3$	9	12	14	17

Table 2. Critical Degrees in F4 for Solving Projected Multivariate systems

B Some Known Formulas Regarding Gröbner basis

We assume to be solving a system with n vars in $\mathbb{F}_q$, m equations; the notation $[u]s$ means the coefficient of the unit or monomial u in the series expansion of s.

Proposition 1 ([1, 36]). *At degree D, the max. number of possible terms in an equation is $N^{(D)} = N = [t^D]\ \dfrac{(1-t^q)^n}{(1-t)^{n+1}}$ which reduces to $\binom{n+D}{D}$ for large q. The number of equations in the elimination stage is $mN^{(D-2)}$ for XL, $N' \approx [t^D]\ \dfrac{(1-t^q)^n}{(1-t)^n}$ for $\mathbf{F_4}$ and $\mathbf{F_5}$. If $\frac{D}{n} = w + o(1)$, then N is exponential in n, or*

$$\begin{aligned}\lg N &\sim n\left[(1+w)\lg(1+w) - w\lg w\right] + o(n), \ \textit{for large } q;\\ &\sim n\left[-(1-w)\lg(1-w) - w\lg w\right] + o(n), \ \textit{over } \mathbb{F}_2;\\ &\sim n\left[\lg\min(z^{-w}(1-z^q)/(1-z))\right] + o(n), \textit{ in general.}\end{aligned}$$

These Gröbner basis methods, sometimes called Lazard-Faugère methods, work for equations of any assorted degree [2, 36]. The principal result is:

Proposition 2 ([2] and [36, Theorem 7]). *Suppose* $\deg p_i := d_i$, *and (*) $\underline{\text{the syzygy module can be generated just from } p_ip_j = p_jp_i \text{ and } p_i^q = p_i}$, then the number of residual degrees of freedom is given by $N - I = [t^D]\ G(t)$, where*

$$G(t) = \frac{(1-t^q)^n}{(1-t)^{n+1}} \prod_{j=1}^{m}\left(\frac{1-t^{d_j}}{1-t^{q\,d_j}}\right), \ \textit{for XL;} \tag{9}$$

$$= \frac{(1-t^q)^n}{(1-t)^{n}} \prod_{j=1}^{m}\left(\frac{1-t^{d_j}}{1-t^{q\,d_j}}\right), \ \textit{for } \mathbf{F_4} \textit{ and } \mathbf{F_5}; \tag{10}$$

Here I means the dimension of the relations generated by our polynomial system up to degree D, taken as a vector space. There is always a certain degree D above which Eq. 9 and hence the underlined condition (*) above cannot continue hold if the system has a solution, because the right hand side of Eq. 9 goes nonpositive. This is $D_{reg} := \min\{D : [t^D]\ G(t) \le 0\}$, called the **degree of regularity**. We sometimes write D_{reg} as D_{XL} (resp. $D_{\mathbf{F_4}}$) for XL (resp. $\mathbf{F_4}/\mathbf{F_5}$). If (*) holds for as long as possible (which means for all $D < D_{reg}$), then $D_0 = D_{reg}$ and we say that the system is **K-semi-regular** or **q-semi-regular** (cf. [1, 36]).

Diem proves [13] for char 0 fields K (and it has been conjectured for all K) that (i) a generic system (no algebraic relationship betweem the coefficients) is K-semi-regular and (ii) if $(p_i)_{i=1\cdots m}$ are *not* K-semi-regular, I can only decrease from the Eq. 9 prediction. Most experts seem to believe the conjecture [13] that a *random* system behaves like a generic system with probability close to 1.

While D_{reg} is still hard to compute, asymptotic analysis is possible. A principal result [2] is that for any given q, as long as m/n is a roughly constant, we know that $D_{reg}/n = w + o(1)$ holds for some constant w.

C Solving via Resultants

Let $Res_i(f, g)$ be the resultant of f and g w.r.t. the variable X_i; ie, the resultant obtained by treating f and g in $\mathbb{L}[X_1, \ldots, X_h]$ as elements of $\mathbb{L}[X_1, \ldots, \widehat{X_i}, \ldots, X_h][X_i]$, where the hat-notation means to skip index i.

To find the common roots of $Q_\ell - C_\ell = 0$, where $\ell = 1, \ldots, h$, first compute $R_i^{(1)} = Res_1(Q_i - C_i, Q_{i+1} - C_{i+1})$ for $i = 1, \ldots, h-1$. As $\deg_{X_1} Q_i \le 2$, each resultants is a determinant of a 4×4 matrix at most, with entries at most quadratic in $X_2, \ldots, X_t$. Thus each $R_i^{(1)} \in \mathbb{L}[X_2, \ldots, X_h]$ is total degree 4. These have the property that if $(A_2, \ldots, A_h)$ is a zero of $R_i^{(1)}$ then $\exists A_1 \in \mathbb{L}$ such that $(A_1, A_2, \ldots, A_h)$ is a common root of $Q_i - C_i$ and $Q_{i+1} - C_{i+1}$.

Now compute $R_i^{(2)} = Res_2(R_i^{(1)}, R_{i+1}^{(1)})$ for $i = 1, \ldots, h-2$. These will be 8×8 determinants and total degree 16 in $X_3, \ldots, X_t$. When $h = 3$, we now have a single univariate polynomial of degree at most 16 which is satisfied by the X_3-coordinate of any solution to our original system. For $h > 3$, we can continue and solve at most by $j = h - 1$. The size of the matrix whose determinant is $R_i^{(j)}$ is 2^{2^j}, and the overall number of determinants is $\frac{h(h-1)}{2}$.

We solve in at most $j = h - 1$ steps. The size of the matrix whose determinant is a result at step j is 2^{2^j}, and the overall number of determinants is $\frac{h(h-1)}{2}$.

We know from results on Gröbner basis (e.g., [2]) methods that the eventual degree of the univariate equation should be at most 2^h. Given that, the above method seems too complicated and for sure will spend a lot of time computing coefficients that eventually cancel. Hence our choice of a tailored Gröbner basis method in Sec. 5.2.