A "Medium-Field" Multivariate Public-Key Encryption Scheme

Lih-Chung Wang[1,⋆], Bo-Yin Yang[2,3,⋆⋆], Yu-Hua Hu[4,⋆], and Feipei Lai[4]

[1] Department of Applied Mathematics, National Donghua University, Hualien, Taiwan
lcwang@math.ndhu.edu.tw
[2] Department of Mathematics, Tamkang University, Tamsui, Taiwan
[3] Taiwan Information Security Center, Taipei
by@moscito.org
[4] Department of Computer Science and Engineering, National Taiwan University, Taipei, Taiwan
{d92015, flai}@csie.ntu.edu.tw

Abstract. Electronic commerce fundamentally requires two different public-key cryptographical primitives, for key agreement and authentication. We present the new encryption scheme MFE, and provide a performance and security review. MFE belongs to the $\mathcal{MQ}$ class, an alternative class of PKCs also termed Polynomial-Based, or multivariate. They depend on multivariate quadratic systems being unsolvable.

The classical trapdoors central to PKC's are modular exponentiation for RSA and discrete logarithms for ElGamal/DSA/ECC. But they are relatively slow and will be obsoleted by the arrival of QC (Quantum Computers). The argument for $\mathcal{MQ}$-schemes is that they are usually faster, and there are no known QC-assisted attacks on them.

There are several $\mathcal{MQ}$ digital signature schemes being investigated today. But encryption (or key exchange schemes) are another story — in fact, only two other $\mathcal{MQ}$-encryption schemes remain unbroken. They are both built along "big-field" lines. In contrast MFE uses medium-sized field extensions, which makes it faster. For security and efficiency, MFE employs an iteratively triangular decryption process which involves rational functions (called by some "tractable rational maps") and taking square roots. We discuss how MFE avoids previously known pitfalls of this genre while addressing its security concerns.

Keywords: multivariate ($\mathcal{MQ}$) public key cryptosystem, Galois field, extended triangular form, tame-like map, tractable rational map, MFE.

⋆ Sponsored in part by National Science Council under Grant NSC-94-2115-M-259-002.
⋆⋆ Correspondence should be addressed to this author, who is partially sponsored by the Taiwan Information Security Center (TWISC) project as well as the National Science Council under Grant NSC-94-2115-M-032-010. A version of this paper is available as a TWISC tech report and will be placed on the IACR ePrint archive.

D. Pointcheval (Ed.): CT-RSA 2006, LNCS 3860, pp. 132–149, 2006.

1 Introduction

Electronic commerce requires at least the following fundamental cryptological primitives: one digital signature scheme, one public-key encryption or key exchange scheme, one hash function, and one symmetric cipher. The first two involve public-key cryptosystems which are based on computationally difficult problems. Currently deployed PKCs most often involve the integer factoring problem (RSA, ESign, Rabin) or the discrete log problem (ECC, ElGamal/DSA).

We aim to introduce a new public-key encryption scheme that may be used for key exchange. This is one of many schemes based on the difficulty of solving a system of polynomial equations. PKCs of this class are usually described as $\mathcal{MQ}$, multivariate, or polynomial-based schemes.

Before we proceed, let us first answer the inevitable question of why researching alternative schemes at all when RSA does just fine. One reason is for diversity. When quantum computers that can handle 4000 quantum bits becomes reality, *Shor's Algorithm* [Sho94] can break all the abovementioned classical cryptosystems very quickly. $\mathcal{MQ}$-schemes are among the alternative PKCs that are weakened by quantum computers (via Grover's Algorithm [Gro96]), but not fatally wounded. Another is for better efficiency in resource (time, power or chip area) usage. This can let us do public-key cryptography in low-resource environments, or make do with cheaper components where we already use PKI.

We will introduce $\mathcal{MQ}$-schemes (Sec. 2), and discuss their state of the art in Sec. 3. We construct the central map of our schemes in Sec. 4, and explain our idea based on an overlay of two stepwise triangular systems. Further details are given in Sec. 5 and the Appendices. Aspects of security are sketched in Sec. 6–8, and we conclude with a discussion of performance and future possibilities.

2 About Multivariate or $\mathcal{MQ}$-Schemes

An $\mathcal{MQ}$-scheme is a cryptosystem whose security depend on this problem:

> Solve the system $p_1 = p_2 = \cdots = p_m = 0$, where each p_i is a quadratic polynomial in $x_1, \ldots, x_n$. All coefficients and variables are in $\mathbb{K} = \mathrm{GF}(q)$.

This problem is called $\mathcal{MQ}$ (for multivariate quadratics). The complexity clearly depend on q, the size of the finite field $\mathbb{K}$ (usually called the *base field*). [GJ79] proved $\mathcal{MQ}$ to be generically NP-hard even over the smallest field, i.e., when $q = 2$. Of course, that does not necessarily imply $\mathcal{MQ}$ to be difficult *on average*, but prevailing expert opinion does expect it to be exponential-time.

The public map $\mathcal{P}$ is the set of quadratics $(p_1, \ldots, p_m)$. Of course, we need a trapdoor to build a public-key cryptosystem. In every practical $\mathcal{MQ}$-schemes, this is accomplished by having a $\mathcal{P}$ that is composed of three maps as in $\mathcal{P} = T \circ \mathcal{Q} \circ S$. $\mathcal{Q}$ is the *central map* and it is quadratic. S and T are linear (affine) maps. We can write them as $S : \mathbf{x} \mapsto \mathbf{x}' = \mathrm{M}_S\,\mathbf{x} + \mathbf{c}_S$, $T : \mathbf{y}' \mapsto \mathbf{y} = \mathrm{M}_T\,\mathbf{y}' + \mathbf{c}_T$. Some authors also represent this as $\mathcal{P} : \mathbf{x} \in \mathbb{K}^n \overset{S}{\mapsto} \mathbf{x}' \overset{\mathcal{Q}}{\mapsto} \mathbf{y}' \overset{T}{\mapsto} \mathbf{y} \in \mathbb{K}^m$. We may set $\mathcal{P}(0) = 0$. The public key is then the $mn(n+3)/2$ nonconstant coefficients of

$\mathcal{P}$ (we assume $q > 2$). The secret key comprise the $n(n+1) + m(m+1)$ entries of $(\mathrm{M}_S^{-1}, \mathrm{M}_T^{-1}, \mathbf{c}_S, \mathbf{c}_T)$, plus parameters in $\mathcal{Q}$ needed for taking its inverse.

If $\mathcal{Q}$ is a random quadratic, then $\mathcal{P}$ would be equally random and infeasible to decompose. But that is impossible, since we need to invert $\mathcal{Q}$ efficiently. Thus *the security of an $\mathcal{MQ}$-scheme depends on the infeasibility of decomposing maps in addition to that of solving large systems.*

3 Current $\mathcal{MQ}$-Schemes and Taxonomy

[WP05] is a good reference on the nomenclature and state-of-the-art on $\mathcal{MQ}$-schemes today. According to its classification, known $\mathcal{MQ}$-schemes (extant and broken) are a handful of modifiers applied to four different basic trapdoors (all must be modified in practice) and two combinations:

C^* or MIA: C^* (and HFE below) can be used both for digital signatures and for encryption. Proposed by Matsumoto-Imai [MI88], broken and revamped by Patarin into the signature scheme C^{*-} or MIA− [Pat95, PGC98]. NESSIE-recommended SFLASH [Nessie, PCG01a] is an instance. Ding proposed PMI+ (MIAi+) for encryption [DG05] (a modification from its precursor MIAi [Din04] after the cryptanalysis of Fouque *et al* [FGS05]).

HFE: The basic scheme of Patarin's Hidden Field Equations [Pat96] is broken ([CDF03, FJ03]), but HFE− or HFEv− (or QUARTZ, [PCG01]) for signatures and HFEi (or IPHFE, [DS05]) for encryption are not. This genre of schemes are burdened by its slow private map.

UOV: Unbalanced-Oil-and-Vinegar by Kipnis *et al* [KPG99], a modification of the earlier and broken [KS98] Oil-and-Vinegar. Useful for signing only and secure only for some awkward choices of parameters [BWP05], hence usually appears in combination with STS.

STS: Stepwise Triangular System. Variables are solved one by one in domino fashion. This basic trapdoor and its ± modifications is broken by techniques of [GC00] (also [CSV93, WBP04, YC05]). Two useful combinations follow:

STS-UOV: A better name might be LuOV (*Layered unbalanced Oil-and-Vinegar*). In segments, vinegar variables are added and linear systems are solved *a la* UOV. All extant examples are very new signature schemes. The first is enTTS (Enhanced Tame Transformation Signature, [YC05]), which is a *sparse variant* just like TRMS (Tractable Rational Map Signature, [WHL+05]). In the slightly later Rainbow [DS05a], Ding *et al* decide to omit the sparsity.

STS-R: Stepwise Triangular System Repeated. Iteration of triangular runs are made to cover inevitable rank vulnerabilities in a triangular system.

[WP05] calls MIA/HFE *mixed-field* ([YC05] terms them *big-field*). These are run mostly over one single large field even though the public map is over a smaller field. STS/UOV is in contrast called *single-field* or *true* [WP05, YC05] because we actually work with the field units. Note that in TRMS several field extensions are used at once. But if we expand all products, we see that they only serve to create an efficiently-invertible sparse central map.

4 A Central Map for an $\mathcal{MQ}$-Scheme, on Medium Fields

We now describe our idea, a central map $\mathcal{Q}$ with a very different flavor:

1. We use throughout the private map one particular field extension $\mathbb{L}$ above the base field $\mathbb{K}$. But this field extension does not cover (nearly) all the variables. Hence, we are not dealing with a *big-field* $\mathcal{MQ}$-scheme like in MIA or HFE. Hence the title "Medium Field" and the name MFE.
2. We are solving for the variables in stepwise fashion, *without using vinegar variables*. Indeed, our scheme might be said to descend spiritually from TTM [Moh99], which pioneered the STS-R approach, to combine two triangular maps to cover the critical small end of the triangle. However, all implementations of TTM had fatal flaws, and our techniques and ideas are radically different, involving what have been called "tractable rational maps" [WC04].
3. Our intended as practical example will have a base field of $\mathbb{K} = \mathrm{GF}(2^{16})$. While it is not unknown for multivariates to have such a base field, it is usually the result of scaling up for security reasons. Here we are designing from the ground up to use such a big field.
4. Our currently favored example scheme is also *tame-like* [YC05]. This makes the key generation process more efficient.

4.1 The Central Map

We define $\mathcal{Q} : \mathbb{L}^{12} \to \mathbb{L}^{15}$ as follows:

$$\begin{cases} Y_1 = X_1 + X_5X_8 + X_6X_7 + Q_1; \\ Y_2 = X_2 + X_9X_{12} + X_{10}X_{11} + Q_2; \\ Y_3 = X_3 + X_1X_4 + X_2X_3 + Q_3; \\ Y_4 = X_1X_5 + X_2X_7; & Y_5 = X_1X_6 + X_2X_8; \\ Y_6 = X_3X_5 + X_4X_7; & Y_7 = X_3X_6 + X_4X_8; \\ Y_8 = X_1X_9 + X_2X_{11}; & Y_9 = X_1X_{10} + X_2X_{12}; \\ Y_{10} = X_3X_9 + X_4X_{11}; & Y_{11} = X_3X_{10} + X_4X_{12}; \\ Y_{12} = X_5X_7 + X_2X_{11}; & Y_{13} = X_5X_{10} + X_7X_{12}; \\ Y_{14} = X_6X_9 + X_8X_{11}; & Y_{15} = X_6X_{10} + X_8X_{12}. \end{cases} \tag{1}$$

Here each X_i and Y_i is in $\mathbb{L} = \mathbb{K}^k$. Since $\mathbb{L} = \mathbb{K}^k$. We split $X_1, X_2, X_3, Q_1, Q_2, Q_3$ into components in $\mathbb{K}^k$, such that $q_1 = 0$, $q_2 = (x'_1)^2$ and for $i = 3 \cdots 3k$, q_i is a more or less a random quadratic in variables $(x'_1, \ldots, x'_{i-1})$.

$$X_1 = \begin{bmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_k \end{bmatrix}, X_2 = \begin{bmatrix} x'_{k+1} \\ x'_{k+2} \\ \vdots \\ x'_{2k} \end{bmatrix}, X_3 = \begin{bmatrix} x'_{2k+1} \\ x'_{2k+2} \\ \vdots \\ x'_{3k} \end{bmatrix}; Q_1 = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_k \end{bmatrix}, Q_2 = \begin{bmatrix} q_{k+1} \\ q_{k+2} \\ \vdots \\ q_{2k} \end{bmatrix}, Q_3 = \begin{bmatrix} q_{2k+1} \\ q_{2k+2} \\ \vdots \\ q_{3k} \end{bmatrix}.$$

4.2 An Inverse to the Central Map

Idea: We may arrange $X_1, X_2, \ldots, X_{12}, Y_4, Y_5, \ldots, Y_{15} \in \mathbb{L}$, into 2×2 matrices:

$$M_1 = \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix}, \quad M_2 = \begin{bmatrix} X_5 & X_6 \\ X_7 & X_8 \end{bmatrix}, \quad M_3 = \begin{bmatrix} X_9 & X_{10} \\ X_{11} & X_{12} \end{bmatrix}; \tag{2}$$
$$M_1 M_2 = \begin{bmatrix} Y_4 & Y_5 \\ Y_6 & Y_7 \end{bmatrix}, \; M_1 M_3 = \begin{bmatrix} Y_8 & Y_9 \\ Y_{10} & Y_{11} \end{bmatrix}, \; M_2^T M_3 = \begin{bmatrix} Y_{12} & Y_{13} \\ Y_{14} & Y_{15} \end{bmatrix}.$$

$\mathcal{Q}$ is inverted in three triangular steps, simple linear algebra gives that

$$Y_4 Y_7 - Y_5 Y_6 = \det(M_1 M_2) = \det M_1 \det M_2,$$

and similarly,

$$Y_8 Y_{11} - Y_9 Y_{10} = \det M_1 \det M_3, \; Y_{12} Y_{15} - Y_{13} Y_{14} = \det M_2 \det M_3.$$

Thus, knowing $Y_4, \ldots, Y_{15}$, we can find $\det M_1$, $\det M_2$, and $\det M_3$, provided that none of them is zero (we will need a square-root taking operation, which is one-to-one and onto, and not very hard – as we shall show – in a char $= 2$ field, and for appropriately chosen k). Further

$$Y_1 = X_1 + \det M_2 + Q_1, \; Y_2 = X_2 + \det M_1 + Q_2, \; Y_3 = X_3 + \det M_3 + Q_3.$$

Therefore, having found $\det M_1, \det M_2, \det M_3$, we reduce the components of Y_1, Y_2, Y_3 to a triangular form in the x_i:

$$X_1 + Q_1 = Y_1 + \sqrt{(Y_4 \times Y_7 + Y_5 \times Y_6)(Y_8 \times Y_{11} + Y_9 \times Y_{10})(Y_{12} \times Y_{15} + Y_{13} \times Y_{14})^{-1}}$$
$$X_2 + Q_2 = Y_2 + \sqrt{(Y_4 \times Y_7 + Y_5 \times Y_6)(Y_8 \times Y_{11} + Y_9 \times Y_{10})^{-1}(Y_{12} \times Y_{15} + Y_{13} \times Y_{14})}$$
$$X_3 + Q_3 = Y_3 + \sqrt{(Y_4 \times Y_7 + Y_5 \times Y_6)^{-1}(Y_8 \times Y_{11} + Y_9 \times Y_{10})(Y_{12} \times Y_{15} + Y_{13} \times Y_{14})}$$

then we apply a second triangular step to compute X_1, X_2, and X_3 component by component. If $X_1 \neq 0$, from $\det M_1$ we can also find X_4. We can now obtain the rest of the variables. **However, it is not necessary to have $X_1 \neq 0$.** This will make a difference in the security analysis — we omit the details, please see Appendix B.

It remains to flesh out our skeletal description, explain our design decisions, and try to show how our approach avoids the mistakes made in earlier designs.

5 Sample Implementations Using a "Tower" Approach

We start by taking $\mathbb{L} = \mathbb{K}^4$. We will use (for simplicity) $q_i = c_i x_{i-1} x_{i-2}$ for $i = 3 \cdots 11$. $\mathcal{S}$ and $\mathcal{T}$ are respectively affine maps of $\mathbb{L}^{12}$ and $\mathbb{L}^{15}$, selected according to Eq. 5 such that the resultant public map $\mathcal{P} = \mathcal{T} \circ \mathcal{Q} \circ \mathcal{S}$ does not have a constant term.

5.1 The Sample Scheme MFE-1

We will take $\mathbb{K} = \mathrm{GF}(2^{16})$. $\mathbb{K}$ is implemented as a degree-two extension of $\mathrm{GF}(2^8)$, and $\mathbb{L}$ as a degree four extension of $\mathbb{K}$ that is a composition of two degree-two extensions. We multiply in $\mathrm{GF}(2^8)$ via a 256×256 (64kB) table. We may alternatively use log-and-exp tables, which are somewhat slower.

From $\mathrm{GF}(2^8)$ to $\mathbb{L}$ we need to do three degree-two extensions. This can be aided by the following observation: Let $F' = F[x]/(x^2 + x + \alpha)$ be a valid deg-2 extension of the char-2 finite field F, then so is $F'' = F'[y]/(y^2 + y + \alpha x)$ a valid extension of F'. The arithmetic of the "tower" extensions may then be done efficiently using Karatsuba multiplication/inversion ([KO63]):

$$(ax + b)(cx + d) = [(a + b)(c + d) + bd] + [\alpha ac + bd]\,, \tag{3}$$

$$\text{similarly,} \qquad (ax + b)^{-1} = \left[b(a + b) + \alpha a^2\right]^{-1} [ax + (a + b)]\,; \tag{4}$$

where we are operating in a field extension from F to $F[x]/(x^2 + x + \alpha)$. Multiplication by α is a lot easier than a regular multiplication because it is fixed. Squaring is also easier than a regular multiplication. Indeed, we find a normal basis of $\mathrm{GF}(2^8)$, i.e., an x such that $(x, x^2, x^4, x^8, x^{16}, x^{32}, x^{64}, x^{128})$ forms a basis of $\mathrm{GF}(2^8)$, and represent by a byte $\sum_{i=0}^{7} b_i 2^i$ the element $\sum_{i=0}^{7} b_i x^{2^i} \in \mathrm{GF}(2^8)$. Build a multiplication (or log-exp) table accordingly. Squaring then become no more than a byte rotation. It is consistent with our own implementations that *with every extra stage in the tower, a multiplication or division takes a little more than* $3\times$ *time. A squaring always cost less than* $1/10$ *of a multiplication.*

Keys and Generation: The private key is the coefficients in S, T and the c_i's for a total of 5,904 elements of $\mathbb{K}$ or about 12kB. The public key comprise the $60 \times 48 \times 51/2 = 73,440$ coefficients of $\mathcal{P}$ or about 147kB.

While $\mathcal{MQ}$-schemes typically use interpolation or a similar technique (cf. [Wol04]) to generate the key, **a faster method [YC05, YCCh04] applies here because our scheme is tame-like** [YC05]. Let $\mathcal{P}$ be given by the quadratics

$$y_k = \sum_i P_{ik} x_i + \sum_i Q_{ik} x_i^2 + \sum_{i>j} R_{ijk} x_i x_j,\ k = 1 \cdots m.$$

Expand each central equation in Y_i into its 4 y'_i components. Each central equation in y'_γ has less than 20 cross-terms $\pi_{\alpha\beta} x'_\alpha x'_\beta$, where $\pi_{\alpha\beta}$'s are constants of the system or one of the c_i's (we keep these metadata in a precomputed table).

$$P_{ik} = \sum_{\gamma=1}^{m} \left[(\mathrm{M}_T)_{k,\gamma} \left([\gamma \le 12](\mathrm{M}_S)_{\gamma i} + \sum_{\pi_{\alpha\beta}\ x'_\alpha x'_\beta \text{ in } y'_\gamma} \pi_{\alpha\beta} \left((\mathrm{M}_S)_{\alpha i} (\mathbf{c}_S)_\beta + (\mathbf{c}_S)_\alpha (\mathrm{M}_S)_{\beta i} \right) \right) \right]$$

$$Q_{ik} = \sum_{\gamma=1}^{m} \left[(\mathrm{M}_T)_{k,\gamma} \left(\sum_{\pi_{\alpha\beta}\ x'_\alpha x'_\beta \text{ in } y'_\gamma} \pi_{\alpha\beta} (\mathrm{M}_S)_{\alpha i} (\mathrm{M}_S)_{\beta i} \right) \right]$$

$$R_{ijk} = \sum_{\gamma=1}^{m} \left[(\mathrm{M}_T)_{k,\gamma} \left(\sum_{\pi_{\alpha\beta}\, x'_\alpha x'_\beta \text{ in } y'_\gamma} \pi_{\alpha\beta} \left((\mathrm{M}_S)_{\alpha i}(\mathrm{M}_S)_{\beta j} + (\mathrm{M}_S)_{\alpha j}(\mathrm{M}_S)_{\beta i} \right) \right) \right]$$

In the formula for P_{ik}, the notation $[\gamma \leq 12]$ means a term that is only present if $\gamma \leq 12$. For every pair $i < j$, we first find

$$\overline{R}_{ijk} = \sum_{\pi x_\alpha x_\beta \text{ is a term of } y_k} \left[\pi \left((\mathrm{M}_S)_{\alpha i}(\mathrm{M}_S)_{\beta j} + (\mathrm{M}_S)_{\alpha j}(\mathrm{M}_S)_{\beta i} \right) \right]$$

for every k, then multiply the vector by the matrix M_T to find all R_{ijk} at once ([YC05, YCCh04]). We are then able to compute the entire key in less than 5 million $\mathbb{K}$-multiplications, Finally, the constant part $\mathbf{c}_T$ of $\mathcal{T}$ is computed thus:

$$(\mathbf{c}_T)_k = \sum_{p=1}^{n} \left[(\mathrm{M}_T)_{k,\ell} \left((\mathbf{c}_S)_\ell + \sum_{\pi\, x_\alpha x_\beta \text{ in } y_\ell} \pi\, (\mathbf{c}_S)_\alpha (\mathbf{c}_S)_\beta \right) \right]. \tag{5}$$

Details of encryption and decryption operations may be filled in as above.

5.2 Other Sample Schemes

Aside from the "regular" scheme with $\mathbb{K} = \mathrm{GF}(2^{16})$ and $k = 4$. We will also present contrasting data for other instances of MFE.

MFE-1′: Here we use $k = 5$ ($\mathbb{L} = \mathrm{GF}(2^{80})$) instead of 4. The computations are significantly more complex and time-consuming.

MFE-0: In what we shall call the "mini"-implementation, we use $\mathbb{K} = \mathrm{GF}(2^8)$ and $\mathbb{L} = \mathrm{GF}(2^{32})$. Everything else is as in the above section.

MFE-0$^+$: Run like MFE-0, but we use redundancy to make failure to decrypt less likely. When encrypting, always treat two blocks (B, B') by sending $\mathcal{P}(B), \mathcal{P}(B'), \mathcal{P}(B' \boxplus B^{-1})$, where B^{-1} means to take the patched inverse in $\mathbb{K}$ of every component, and $\boxplus$ means addition modulo $|\mathbb{K}|$.

$\mathrm{GF}(2^8)$ multiplications are three times faster than $\mathrm{GF}(2^{16})$ multiplications. Therefore, running the encryption function of MFE-0 three times is still faster than one run of the encryption function of MFE-1, and let us have the smaller key sizes of MFE-0. Other implications are given below.

6 Security: A Basic Overview

Security analysis for $\mathcal{MQ}$-schemes are hampered by the lack of "provable security". As far as we know, the only attempt in this area is due to [Cou03], which is not followed up actively. As a result, while it is easy to show that an $\mathcal{MQ}$-scheme is insecure by presenting a cryptanalysis. So far a cryptologist can only show that current attacks don't work and are not likely to work. We try to do the best we can under the circumstances. Currently known attacks on polynomial-based PKCs can be roughly classified into four kinds:

Correlation/Statistical Attacks: A common systemic attacks agains symmetric ciphers, but usually not applied against PKCs. We will describe how known attacks do not apply.

Linear Algebra-Based (Rank) Attacks: There are several attacks that are quite generic against when the target scheme is not of the big-field type. We just give numbers below, please refer to Appendix.

Algebraic Attacks: Today this means any attack whose functionality comes down to solving a system of equations, usually distilled out of the structure of the system. We will summarize what is out there.

Very Specialized Attacks: Obviously, an attacks has to focus on some aspect or structural element of the cryptosystem. Some do have wide applicability to a whole class of schemes. Others do not. While this does not detract from the sheer intellectual worth or ingenuity of such cryptanalytic work, many ideas (e.g., the Gilbert-Minier attack [GM02] on the original SFLASH) simply do not work on other schemes.

We repeat that showing our scheme to resist known attacks does not guarantee security. It is an "original sin" for $\mathcal{MQ}$-schemes today. We only do what we can.

6.1 Cryptanalysis Using Rank Attacks

"Rank Attacks" encompasses the High Rank, Low Rank, and Separation of Oil-and-Vinegar attacks. These are basic attacks against the STS or UOV based trapdoors in $\mathcal{MQ}$-schemes. *Summary: Rank Attacks do not work on any of our sample encryption schemes.*

Separation of Oil and Vinegar (UOV) Attacks: Security level for the "mini" version (MFE-0) is about 2^{100}; the "full" version (MFE-1) is about 2^{140}.

High Rank (Dual Rank) Attack: Security level is about 2^{128} for the "mini" version (MFE-0) and 2^{181} for the full-version (MFE-1).

Low Rank (Rank or MinRank) Attack: Security level is about 2^{128} for the "mini" version (MFE-0) and 2^{172} for the full-version (MFE-1).

See Appendix A and references, particularly [GC00, YC05] for more details.

6.2 Cryptanalysis Under Specialized Attacks

We tested our scheme not to succumb to any earlier known special attacks, and will henceforth ignore specialized attacks without wider applicability.

One specialized attack that is specifically designed around is the Patarin Relations, which can also be considered an algebraic attack or a linear-algebra attack. It can defeat many systems that can be described as of intrinsic rank 2.

Eq. 2 specifically had its matrix products arranged M_1M_2, M_1M_3 and $M_2^TM_3$. This take full advantage of the incommutativity of matrix multiplication. Other arrangements will create lots of Patarin relations. For example, if we have $N = M_1M_2$ and $N' = M_2M_3$ as matrices with components linear in the Y_i's, then we will have the relations corresponding to $NM_3 = M_1N'$. As the central equations Eq. 2 are written, no such Patarin relations can be found. We should only need to test this over GF(2); we actually tested it over GF(4).

7 Algebraic Cryptanalysis

Basically, an algebraic attack refers to any technique that ends with a system-solving exercise. There may be guessing, or there may not be. The system may be linear, as in Patarin relations vs. C^* [Pat95], or non-linear, as in the Courtois/Faugère-Joux attack on HFE [CDF03, FJ03].

As characterized above, eventually an algebraic attack comes down to solving a system. At the moment, the state of the art is represented by the $\mathbf{F_5}$ Gröbner Bases algorithm of J.-C. Faugère [Fau02] while the best in commercially available software is a version of its predecessor $\mathbf{F_4}$ [Fau99]. Few know how to program $\mathbf{F_5}$ correctly, and certainly the only known implementation of $\mathbf{F_5}$ is the one used by Dr. Faugère to break HFE challenge 1.

The alternative methods of XL/FXL by Courtois *et al* [CKP+00] has been analyzed in some depth [Die04, YC04]. Today XL is usually considered to be a poor relative of $\mathbf{F_4}$-$\mathbf{F_5}$ [AFS+04]. The asymptotic behavior of the Gröbner Bases-XL family is described by [BFS04, BFS+05, YCCo04].

7.1 On Extraction of More Tractable Systems

An important remark is that the attack against HFE challenge 1 involves an algebraic extraction of the actual system to be solved, one that is significantly more overdetermined than the original.

Such an extraction method does not yet exist in general. At the moment, we have no way of distinguishing our encryption scheme from random quadratics. This conclusion is supported by some experiments that we ran, trying to solve systems directly very miniaturized version of MFE, using GF(4) as the base field, and the tool is MAGMA and the version of $\mathbf{F_4}$ built therein.

Finally, we cannot rule out the possibility that a method of extracting some solvable system exists, as for example in [JKM+05]. However, we have checked for it and as far as we can tell, no known method of such extraction works.

Let us describe the [JKM+05] attack briefly. This is an attack on the encryption scheme TRMCv2 of Wang and Chang [WC04]. The scheme in question has many variables and equations, but there is a subsystem of 7 variables and 11 equations. This was part of the trapdoor in TRMCv2. The weakness is that by running a simplified version of XL, the attacker can essentially isolate this central subsystem. It transpires that the algorithm terminates at the degree that is required to solve the central subsystem, instead of the much higher degree that would be required of a generic system with as many equations and variables.

In addition to trying to execute the attack of [JKM+05] and determining that it does not work at a sufficiently low degree, we ran a clique-finding algorithm on our central polynomials and found no such central subsystem.

7.2 On the Speed of Equation-Solving

It seems that the most effective XL-derived method is FXL [YCCo04], and the same idea applies to $\mathbf{F_5}$ at least in the generic case. There are also cases [YC04] where XL will not and $\mathbf{F_5}$ may not work.

Indeed, our formulation satisfies a lemma from [YC04] which says that XL will not work and FXL will take longer. $\mathbf{F_5}$ is also expected to take longer. However, in the following we still assume that FXL and $\mathbf{F_5}$ will function as if the polynomials are generic.

Assuming generic equations, $\mathbf{F_4}$-$\mathbf{F_5}$ works at the smallest degree D where the coefficient of t^D in $(1-t)^{m-n}(1+t)^m$ is negative ([BFS+05, Die04, YC04]). The dominant term of the time complexity is given by $E\left(\binom{n+D-1}{D}\right)$, where $E(N)$ is the cost of elimination on a $N \times N$ matrix equation. Here we have 48 variables and 60 equations. Assuming a field size of $q = 256$ (our "mini" version MFE-0), and $E(N) = N^2(4 + \lg N/4)$ in cycles (a very optimistic assumption that dense-matrix elimination can work asymptotically like sparse-matrix system solving that we take from [YCCo04]), we get about 2^{97} cycles or about a 2^{93} multiplications security level.

The conclusion is, then, that even discounting the possibility that such methods don't function at all, Gröbner Bases and related methods should be more effective than Rank Attacks against our schemes, but does not reduce them down below 2^{80}, with a lot of safety margin.

8 Correlation or Statistical Cryptanalysis and Defenses

A correlation or statistical attack works by finding imbalances of some kind in the ciphertext. Not many attacks on public-key cryptosystems use correlation or statistical artifacts. We comment on only two particular items specifically.

One is the attack [FGS05] on the scheme PMI. Fouque *et al* used differential cryptanalysis with a one-sided statistical distinguisher. While ingenious, this does not apply to our scheme and this is confirmed by some empirical tests.

The other is much more relevant. Our system requires $X_1X_4 - X_2X_3$, $X_5X_8 - X_6X_7$, and $X_9X_{12} - X_{10}X_{11}$ all to be non-zero to get a successful decryption. Thus a single block will fail to decrypt with the probability $3/|\mathbb{L}|$.

For the "mini" sample scheme MFE-0, this chance of failure is about 2^{-30} and for the "regular" sample scheme MFE-1, it is about 2^{-62}. This results in a possible attack by guessing at decryption failures. However, there are no ways an attacker can easily check that two decryption faults correspond to a zero in the same determinant. Nor can we generate easily from two samples where $X_1X_4 = X_2X_3$ another such point to make use of the algebraic variety. We can not guarantee that this takes care of *all* correlation attacks based on decryption faults. However, we can expect any such attack to be significantly more difficult than just finding one such inscrutable ciphertext. Therefore, we can presume that our "regular" schemes are quite secure enough under such attacks.

8.1 Possible Cryptanalysis Via Correlation and Timings

Note that $X_1 = 0$ is much more useful to an attacker than any of the determinants being zero, since it is an affine formula. An adversary that can distinguish whether $X_1 = 0$ can execute the following attack:

Cryptanalysis: encrypt random blocks (vectors) and send to a decryption device (or oracle) for decryption continuously; register the blocks B_i whose corresponding ciphertext result in decryption failures (or timing "tells"). Every time a block B_i is registered, send $aB_i + (1-a)B_j$ for decryption for a few random a's and for each $j < i$ to find out if it correspond to a vector where $X_1 = 0$ (because such vectors form an affine subspace). Collect 12 such blocks in an expected $12|\mathbb{L}|$ attempts. With high probability we have found the affine subspace $X_1 = 0$. Restrict the polynomials to this affine subspace, and we can perform a MinRank attack on the reduced equations corresponding to Y_3. Since each evaluation takes about $mn(n+3)/2$ multiplications, total time used is $12|\mathbb{L}| \cdot [mn(n+3)/2\mathbb{K}\text{-multiplications} + \text{decryption time}]$.

This idea of using decryption failures seems to have been proposed by Proos *et al* [HNP+03]. The elegant cryptanalysis proposed in that paper resulted in a revision in the current version of NTRU Encryption. The idea of using MinRank after some other reduction may have been invented by J. Ding against the predecessor version to enTTS [DY04].

The analysis above shows that the "mini" scheme MFE-0 cannot be used if a Proos-like attack can execute. However, all is not lost. According to Appendix B we can decrypt even when $X_1 = 0$ without an appreciable speed difference, and the same straightforward attack cannot function with $X_1X_4 - X_2X_3 = 0$, say, because it is not an affine relation. Therefore, the method in Appendix B guarantees that such a cryptanalysis will not operate.

Further, for MFE-1, we have $|\mathbb{L}| = 2^{64}$, $n = 48$, $m = 60$, we find the complexity of the cryptanalysis to be about 2^{85} multiplications in $\mathbb{L}$, which is just about barely enough even if a good distinguisher exists for a Proos-like attack. For MFE-1′, where $\mathbb{L} = \mathbb{K}^5$ rather than $\mathbb{L} = \mathbb{K}^4$, the system runs quite a bit slower but the Proos-style attack will still have a cryptanalytic complexity above 2^{90} even if it works. Finally, Even if we can find an alternative way to execute the Proos attack on MFE-0, we can still use MFE-0^+.

9 Performance Data

Having shown that our schemes safe under known attacks, it then becomes meaningful to test performances. We compare our first implementations with the Crypto++ library (benchmarks at `www.eskimo.com/~weidai/benchmarks.html`).

We wrote our programs in plain C. The Crypto++ libraries are of course highly optimized binaries. We think that the data above shows that the schemes we propose are competitive with RSA and ECC. Of course, Crypto++ is not nearly as well optimized for ECC as for RSA.

For comparison's sake, we recompiled our programs in C51 and tested for performance on a 8051. One block decryption of MFE-1 ($\mathbb{K} = \mathrm{GF}(2^{16})$, $\mathbb{L} = \mathbb{K}^4$) can run on the following smart card development platform (24kB EEPROM, including private key and code; 256 byte **idata**, 10 MHz basic Intel 8052, no extra RAM) in 0.28s. MFE-0 or MFE-0^+, if applicable, will be even faster.

Table 1. Our Schemes on a 1.6GHz Opteron compared with Crypto++ library

Scheme	BlockLen	PublKey	SecrKey	Genkey	SecrMap	PublMap
RSA-1024	1024 bits	128 B	320 B	0.86 sec	4.75 ms	0.18 ms
RSA-2048	2048 bits	256 B	640 B	2.71 sec	28.13 ms	0.45 ms
ECIES-155	310 bits	40 B	20 B	0.02 ms	7.91 ms	12.09 ms
MFE-1	512 bits	12 kB	147 kB	9.90 ms	32 μs	0.86 ms
MFE-1′	640 bits	18 kB	283 kB	23.40 ms	48 μs	1.79 ms
MFE-0	256 bits	6 kB	73 kB	2.21 ms	2.3 μs	0.12 ms
MFE-0$^+$	512 bits	6 kB	73 kB	2.21 ms	7.0 μs	0.39 ms

Given that RSA-1024 on an Infineon SLE-66X64-2P (a costly card with 208 kB ROM, 5052 bytes RAM, 64 kB EEPROM, and 1100-bit Advanced Crypto Engine) takes 0.4s at the same clock, this shows that the idea is of particular interest for situations where resources are scarce. This continues the trend of [ACD+03, YCCh04], that is, multivariates are worth investigating for low-resource and pervasive cryptography even without the interest of diversity.

We also note that usually decrypting is centralized at the servers while encrypting is done by those the masses sending data to the servers. So decrypting is more likely to be resource-intensive. Conversely, someone who uses a smart card to verify his or her identity is more likely to want to receive sensitive data, so a smart card (a low resource item) is more likely to want to do decryption.

A lot of optimizations remains to be done for a new scheme, of course, in particular the degree-five extension may be implemented better. We will of course pursue this direction in the future.

10 Discussions and Summary

We make a few comments about history and speculate on the future.

10.1 A Little History

This is the triangular (or tame, or *de Jonquiere*) map of algebraic geometry:

$$y_1 = x_1,\ y_2 = x_2 + f_2(x_1),\ y_3 = x_3 + f_3(x_1, x_2), \ldots,\ y_n = x_n + f_n(x_1, \ldots,\ x_{n-1}).$$

A PKC based on a triangular central map is known to be weak early on. The idea of using a composition of triangular maps to cover the vulnerability (Segmentwise Triangular System Repeated, STS-R) is pioneered by TTM [Moh99]. However, the execution was faulty and no rank-safe instances are available [GC00, YC05].

10.2 An Issue of Terminology

The seminal idea for our scheme is invented by L.-C. Wang [WC05]. The basic approach is not limited to that of the original TTM. It is more versatile, e.g., in a char = 2 field Eq. 2 may start like this without being really different:

$$y'_1 := [Y_1]_1 = (x'_1)^2 + [\det M_2]_1\,.$$

This type of map is called the "Tractable Rational Map" according to Wang *et al* [WC04, WC05], who term PKCs that uses compositions of "tractable rational maps" as "tractable rational map cryptosystems" (TRMC) [WC04, WC05]. Of course, there is a fine line between being nicely general and overly broad. For example, the central maps of HFE and C^* are both "tractable rational" maps (just as they are "tame transformation" maps, which themselves are a subsets of "tractable rational" maps; it must be added that the authors of TRMC do not claim HFE as a TRMC). How to demarcate clearly between various cryptosystems has not been agreed by all scholars of the $\mathcal{MQ}$ genre. An interested reader can look up what is claimed as "Tractable Rational Map Cryptosystem" in [WC04, WC05] as well as the eponymous pending patent application.

11 Conclusion

No other current instances of multivariates with the STS-R structure, such as TTM (Tame Transformation Method), are being employed today. We think we have shown that while the devil is in the details, there is some merit to the idea of STS-R, in particularly. However, to make it useful, we have to generalize by using more general operations than TTM. In particular, we need introduce rational operations and square roots. Therefore it is not TTM any more.

The introduction of new tricks naturally may introduce vulnerabilities. We showed how the failure of decryption may be enough of a discrepancy for cryptanalysis as Proos *et al* did for the previous version of NTRU. We also show our attempts at avoiding a similar fate. We leave to future historians to judge what and how our scheme will be considered.

As is lamented in many discussions (e.g., [WP05]) about $\mathcal{MQ}$-schemes, some measure of provable security seems to be hard to come by. Many scholars are studying this topic. But even so, we hope to have shown that there is some life in $\mathcal{MQ}$-schemes, in particular non-big-field types.

References

[ACD+03] M. Akkar, N. Courtois, R. Duteuil, and L. Goubin, *A Fast and Secure Implementation of SFLASH*, PKC 2003, LNCS v. 2567, p. 267–278.

[AFS+04] G. Ars, J.-C. Faugère, M. Sugita, M. Kawazoe, and H. Imai, *Comparison of XL and Gröbner Bases Algorithms over Finite Fields*. Asiacrypt 2004, LNCS v. 3329, p. 323–337.

[BFS04] M. Bardet, J.-C. Faugère, and B. Salvy, *Complexity of Gröbner Basis Computations for Regular Overdetermined Systems*, INRIA report RR-5049, and presentation at the ICSPP conference honoring Daniel Lazard.

[BFS+05] M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang, *Asymptotic Behaviour of the Degree of Regularity of Semi-Regular Polynomial Systems*, presentation at the MEGA 2005 conference and a chapter of Ph.D. thesis by M. Bardet, 2004.

[BWP05] A. Braeken, C. Wolf, and B. Preneel, *A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes*, CT-RSA 2005, LNCS v. 3376, p. 29-43.

[CSV93] D. Coppersmith, J. Stern, and S. Vaudenay, *Attacks on the Birational Permutation Signature Schemes*, Crypto 1993, LNCS v. 773, p. 435–443.

[Cou03] N. Courtois, *Generic Attacks and the Security of Quartz*, PKC 2003, LNCS v. 2567, p. 351–364. Also see E-Print Archive 2004/143.

[CDF03] N. Courtois, M. Daum, and P. Felke, *On the Security of HFE, HFEv-, and Quartz*, PKC 2003, LNCS v. 2567, p. 337–350.

[CKP+00] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, Eurocrypt 2000, LNCS v. 1807, p. 392–407.

[Die04] C. Diem, *The XL-algorithm and a conjecture from commutative algebra*, Asiacrypt 2004, LNCS v. 3329, p. 338–353.

[Din04] J. Ding, *A New Variant of the Matsumoto-Imai Cryptosystem through Perturbation*, PKC 2004, LNCS v. 2947, p. 305–318.

[DG05] J. Ding and J. Gower. *Inoculating Multivariate Schemes Against Differential Attacks*, private communication and manuscript, E-Print Archive, 2005/255.

[DS05] J. Ding and D. Schmidt, *Cryptanalysis of HFEv and Internal Perturbation of HFE*, PKC 2005, LNCS v. 3386, p. 288–301.

[DS05a] J. Ding and D. Schmidt, *Rainbow, a new Digitial Multivariate Signature Scheme*, ACNS 2005, LNCS v. 3531, p. 164–175.

[DY04] J. Ding and Y. Yin, *Cryptanalysis of a TTS Implementation*, presentation at the IWAP 2004 conference.

[Fau99] J.-C. Faugère, *A New Efficient Algorithm for Computing Gröbner Bases (F4)*, Journal of Pure and Applied Algebra, 139 (1999), p. 61–88.

[Fau02] J.-C. Faugère, *A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)*, Proc. ISSAC, ACM Press, 2002.

[FJ03] J.-C. Faugère and A. Joux, *Algebraic Cryptanalysis of Hidden Field Equations (HFE) Cryptosystems Using Gröbner Bases*, Crypto 2003, LNCS v. 2729, p. 44-60.

[FGS05] P.-A. Fouque, L. Granboulan, and J. Stern, *Differential Cryptanalysis for Multivariate Schemes*, Eurocrypt 2005, LNCS v. 3494, p. 341–353.

[GJ79] M. Garey and D. Johnson, *Computers and Intractability, A Guide to the Theory of NP-completeness*, Freeman and Co., 1979, p. 251.

[GM02] H. Gilbert and M. Minier, *Cryptanalysis of SFLASH*, Eurocrypt 2002, LNCS v. 2332, pp. 288–298.

[GC00] L. Goubin and N. Courtois, *Cryptanalysis of the TTM Cryptosystem*, Asiacrypt 2000, LNCS v. 1976, p. 44–57.

[Gro96] L. K. Grover, *A fast quantum mechanical algorithm for database search*, Proc. 28th Annual ACM Symposium on Theory of Computing (1996), p. 212–220.

[HNP+03] N. Howgrave-Graham, P. Nguyen, D. Pointcheval, J. Proos, J. Silverman, A. Singer, and W. Whyte, *The Impact of Decryption Failures on the Security of NTRU decryption*, Crypto 2003, LNCS v. 2729, p. 226–246.

[JKM+05] A. Joux, S. Kunz-Jacques, F. Muller, P.-M. Ricordel, *Cryptanalysis of the Tractable Rational Map Cryptosystem*, PKC 2005, LNCS v. 3386, pp. 258–274.

[KO63] A. Karatsuba and Yu. Ofman, *Multiplication of Many-Digital Numbers by Automatic Computers*, Doklady Akad. Nauk SSSR 145(1962), p. 293-294. Translation in Physics-Doklady 7(1963), p. 595-596.

[KPG99] A. Kipnis, J. Patarin, and L. Goubin, *Unbalanced Oil and Vinegar Signature Schemes*, Crypto 1999, LNCS v. 1592, p. 206–222.

[KS98] A. Kipnis and A. Shamir, *Cryptanalysis of the Oil and Vinegar Signature Scheme*, Crypto 1998, LNCS v. 1462, p. 257–266.

[MI88] T. Matsumoto and H. Imai, *Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption*, Eurocrypt 1988, LNCS v. 330, p. 419–453.

[Moh99] T. Moh, *A Public Key System with Signature and Master Key Functions*, Communications in Algebra, 27 (1999), pp. 2207–2222.

[Nessie] NESSIE project homepage: `http://www.cryptonessie.org`.

[Pat95] J. Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*, Crypto 1995, LNCS v. 963, p. 248–261.

[Pat96] J. Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms*, Eurocrypt 1996, LNCS v. 1070, p. 33–48.

[PGC98] J. Patarin, L. Goubin, and N. Courtois, *C^*_{-+} and HM: Variations Around Two Schemes of T. Matsumoto and H. Imai*, Asiacrypt 1998, LNCS v. 1514, p. 35–49.

[PCG01] J. Patarin, N. Courtois, and L. Goubin, *QUARTZ, 128-Bit Long Digital Signatures*, CT-RSA'01, LNCS v. 2020, p. 282–297. Update available at [Nessie].

[PCG01a] J. Patarin, N. Courtois, and L. Goubin, *FLASH, a Fast Multivariate Signature Algorithm*, CT-RSA 2001, LNCS v. 2020, p. 298–307. Update available at [Nessie].

[Sho94] P. W. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, Proc. 35th Annual Symposium on Foundations of Computer Science (S. Goldwasser, ed.), IEEE Computer Society Press (1994), p. 124-134.

[WC04] L.-C. Wang, and F.-H. Chang, *Tractable Rational Map Cryptosystem*, manuscript, E-Print Archive 2004/046.

[WC05] L.-C. Wang, and F.-H. Chang, *Revision of Tractable Rational Map Cryptosystem*, manuscript, on the E-Print Archive.

[WHL+05] L.-C. Wang, Y.-H. Hu, F.-P. Lai, C.-Y. Chou, and B.-Y. Yang, *Tractable Rational Map Signature*, PKC 2005, LNCS v. 3386, p. 244–257.

[Wol04] C. Wolf, *Efficient Public Key Generation for Multivariate Cryptosystems*, Proc. ERACOM Conference and Workshop on Cryptographic Algorithms and their Uses, July 5-6, 2004, also see E-Print Archive 2003/089.

[WBP04] C. Wolf, A. Braeken, and B. Preneel, *Efficient Cryptanalysis of RSE(2)PKC and RSSE(2)PKC*, SCN '04, LNCS v. 3352, p. 294–309.

[WP05] C. Wolf and B. Preneel, *Taxonomy of Public-Key Schemes based on the Problem of Multivariate Quadratic Equations*, manuscript, E-Print Archive 2005/077.

[WP05a] C. Wolf and B. Preneel, *Superfluous keys in Multivariate Quadratic asymmetric systems*, PKC 2005, LNCS v. 3386, p. 275–287. Extended version at E-Print Archive 2004/361.

[YC04] B.-Y. Yang and J.-M. Chen, *All in the XL Family: Theory and Practice*, ICISC 2004, LNCS v. 3506, p. 67–86.

[YC05] B.-Y. Yang and J.-M. Chen, *Rank Attacks and Defence in Tame-Like Multivariate PKC's*, ACISP 2005, LNCS v. 3574, p. 518–531. Older version at E-Print Archive 2004/061.

[YCCh04] B.-Y. Yang, J.-M. Chen, and Y.-H. Chen, *TTS: High-Speed Signatures from Low-End Smartcards,*, CHES 2004, LNCS v. 3156, p. 371-385.

[YCCo04] B.-Y. Yang, J.-M. Chen, and N. Courtois, *On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis*, ICICS 2004, LNCS v. 3269, p. 401-413.

A A Brief Description of Rank Attacks

We describe the linear algebra based attacks briefly.

Separation of Oil-and-Vinegar: Consider a set of polynomials $p_i(x_1, \ldots, x_n)$ where the set of variables $\{x_1, \ldots, x_n\}$ can be partitioned into disjoint portions $\mathcal{V} \uplus \mathcal{O}$, such that no quadratic term has both factors in the *oil set* $\mathcal{O}$. If we specify each variable in the *vinegar set* $\mathcal{V}$, we can solve for variables in $\mathcal{O}$ as a linear system. This is called a UOV structure.

Kipnis *et al* attacked UOV structures by distilling the *oil subspace* [KPG99, KS98]. If the size of the minimal vinegar set in the central equations is v, then we can find the subspace spanned by the oil variables in $q^{2v-n-1}(n-v)^4$ field multiplications.

We may have to manuever further, but such a distillation usually leads to a cryptanalysis of the scheme. A program to find maximal cliques can verify that if we ignore the q_i terms and work with $\mathbb{L}$, then $m = 15$, $n = 12$, $v = 9$. So even for $\mathbb{L} = \mathrm{GF}(2^{32})$ of the "mini" version (cf. Sec. 5), the security level way above 2^{100}, high enough. This seems reasonable because the Kipnis attack seems more inclined toward signature schemes.

High Rank Attack: We can associate with every quadratic polynomial a symmetric matrix. To be exact $p = \sum_{i \le j} a_{ij} x_i x_j + \sum_i b_i x_i$ corresponds (for char = 2) to $\mathrm{M}_p := [A_{ij}]$, where $A_{ij} = a_{ij}$ if $i < j$, a_{ji} if $i > j$, and 0 if $i = j$. Usually $r = \operatorname{rank} \mathrm{M}_p$, if and only if we may write $\sum_{i \le j} a_{ij} x_i x_j = \sum l_a l_b$ for a minimum independent set of linear forms $l_1, \ldots, l_r$.

Equations in the public key tend to be full rank as are most of their linear combinations. However, when a variable x_i does not appear in a polynomial p, the associated matrix will be singular, i.e., $\operatorname{rank} \mathrm{M}_p < n$. Thus, if some variable appears in only one central equation, for most pairs of public polynomials (p_i, p_j), we can find a linear combination $p_i + \lambda_{ij} p_j$ that is less than full rank. The same goes for linear combinations of $(u+1)$-tuples of public polynomials if a variable appears only in u central equation. A simple and a more algebraic (and complete) implementation of this idea is given by [GC00] and [CSV93] respectively.

All told, this attack costs around $\left(un^2 + \frac{n^3}{6}\right) q^u$ multiplications if all goes correctly. Here, each X_i appears at least in 4 equations, so even for $\mathbb{L} = \mathrm{GF}(2^{32})$ in MFE-0 (our "mini" scheme) in Sec. 5, we have a security level above 2^{128}. It is quite a bit higher for the "regular" scheme MFE-1.

Low Rank Attack: This is approximately dual to the previous attack.

If p has rank r, then a random vector $\mathbf{x}$ satisfy $\mathrm{M}_p \mathbf{x} = 0$ with probability q^{-r}. We guess at $\mathbf{x}$ and try to solve for the linear combination that

is M_p. For encryption schemes $m > n$, so there are too many matrices spanned by those corresponding to the public polynomials. In this case we must guess at $\mathbf{x}_1, \ldots, \mathbf{x}_k$, where $k = \lceil m/n \rceil$. This makes the linear system $\sum \lambda_i \mathrm{M}_{p_i} \mathbf{x}_j = 0$, $j = 1 \cdots k$ in the λ_i overdetermined. If there is a unique linear combination with the minimum rank r, we expect to find it within q^{kr} tries.

This is also known as the MinRank kernel attack. When there are more than one kernel of the same minimal rank that are mostly disjoint, we can do better [YC05]. If there are c such kernels, then we expect to find one within $q^{kr}kmn(m+n)/c$ field multiplications.

Here, $Y_3 = X_3 + \det M_3 + X_1 X_2 + [\dot{:}]$, or rather its first component, corresponds to a single equation with the smallest rank where $k = 2$, $r = 2$, $q = 2^{32}$ (for the "mini" version MFE-0). Thus the formulas of [GC00, YC05] both gives more than 2^{128} as the security level.

Please refer to [GC00] for details on High and Low Rank attacks, [KPG99, BWP05] on the unbalanced Oil and Vinegar scheme, and [YC05] for a recent summary.

B Inverting $\mathcal{Q}$ and Circumventing $X_1 = 0$

Here is the last complete algorithm we implemented.

1. First find X_1, X_2, X_3 in a triangular manner from

$$X_1 + Q_1 = Y_1 + \sqrt{(Y_4 \times Y_7 + Y_5 \times Y_6)(Y_8 \times Y_{11} + Y_9 \times Y_{10})(Y_{12} \times Y_{15} + Y_{13} \times Y_{14})^{-1}}$$
$$X_2 + Q_2 = Y_2 + \sqrt{(Y_4 \times Y_7 + Y_5 \times Y_6)(Y_8 \times Y_{11} + Y_9 \times Y_{10})^{-1}(Y_{12} \times Y_{15} + Y_{13} \times Y_{14})}$$
$$X_3 + Q_3 = Y_3 + \sqrt{(Y_4 \times Y_7 + Y_5 \times Y_6)^{-1}(Y_8 \times Y_{11} + Y_9 \times Y_{10})(Y_{12} \times Y_{15} + Y_{13} \times Y_{14})}$$

   The actual pre-computations are:
   (a) Calculate $\det(M_1 M_2) = Y_4 \times Y_7 + Y_5 \times Y_6$.
   (b) Calculate $\det(M_1 M_3) = Y_8 \times Y_{11} + Y_9 \times Y_{10}$.
   (c) Calculate $\det(M_2^T M_3) = Y_{12} \times Y_{15} + Y_{13} \times Y_{14}$.
   (d) Calculate $\det M_1 = \sqrt{\det(M_1 M_2) \det(M_1 M_3) / \det(M_2^T M_3)}$.
   (e) Calculate $\det M_2 = \det(M_1 M_2) / \det M_1$, $\det M_3 = \det(M_1 M_3) / \det M_1$.
   (f) Calculate $Y_1 + Q_1$, $Y_2 + Q_2$, $Y_3 + Q_3$ and the triangular substitutions.
2. if $X_1 \neq 0$ compute M_1^{-1} and thereby M_2 and M_3, and we are done.
3. if $X_1 = 0$, we let $B = (\det(M_2^T M_3))^{-1}$ and compute $A = X_2^{-1}$, then

$$X_7 = Y_4 A$$
$$X_8 = Y_6 A$$
$$X_{11} = Y_8 A$$
$$X_{12} = Y_9 A$$

$$
\begin{aligned}
X_9 &= \det(M_3)\, B\,(Y_{12}X_8 + Y_{14}X_7) \\
X_{10} &= \det(M_3)\, B\,(Y_{13}X_8 + Y_{15}X_7) \\
X_5 &= \det(M_2)\, B\,(Y_{12}X_{12} + Y_{13}X_{11}) \\
X_6 &= \det(M_2)\, B\,(Y_{14}X_{12} + Y_{15}X_{11}) \\
X_4 &= \det(M_3)\, B\,(Y_6X_6 + Y_7X_5)
\end{aligned}
$$

Note that this avoids trouble if any other variable vanishes! We can also see that this case takes 1 fewer multiplication and 4 fewer additions after a careful count, and should pad the time upwards with some delaying action.