Operating Degrees for XL vs. $\mathrm{F}_4/\mathrm{F}_5$ for Generic $\mathcal{M}Q$ with Number of Equations Linear in that of Variables

Jenny Yuan-Chun Yeh, Chen-Mou Cheng, and Bo-Yin Yang

Academia Sinica, Taipei, Taiwan, {jenny,doug,by}@crypto.tw

Abstract. We discuss the complexity of $\mathcal{M}Q$, or solving multivariate systems of m equations in n variables over the finite field $\mathbb{F}_q$ of q elements. $\mathcal{M}Q$ is an important hard problem in cryptography. In particular, the complexity to solve overdetermined $\mathcal{M}Q$ systems with randomly chosen coefficients when $m = cn$ is related to the provable security of a number of cryptosystems.
In this context there are two basic approaches. One is to use XL ("eXtended Linearization") with the solving step tailored to sparse linear algebra; the other is of the many variations of Jean-Charles Faugère's $\mathrm{F}_4/\mathrm{F}_5$ algorithms.
Although $\mathrm{F}_4/\mathrm{F}_5$ has been the de facto standard in the cryptographic community, it was proposed (Yang-Chen, 2004) that XL with Sparse Solver may be superior in some cases, particularly the generic overdetermined case with $m/n = c{+}o(1)$.

At the Steering Committee Meeting of the Post-Quantum Cryptography workshop in 2008, Johannes Buchmann listed several key research questions to all post-quantum cryptographers present. One problem in $\mathcal{M}Q$-based cryptography, he noted, is "if the difference between the operating degrees of XL(-with-Sparse-Solver) and $\mathrm{F}_4/\mathrm{F}_5$ approaches can be accurately bounded for random systems."

We answer in the affirmative when $m/n = c{+}o(1)$, using Saddle Point analysis:

1. For instances with randomly drawn coefficients, the degrees of operation of XL and $\mathrm{F}_4/\mathrm{F}_5$ has the most pronounced differential in the large-field, "barely overdetermined" ($m{-}n = c$) cases, where the discrepancy is $\propto \sqrt{n}$.
2. In most other types of random systems with $m/n = c + o(1)$, the expected difference in the operating degrees of XL and $\mathrm{F}_4/\mathrm{F}_5$ is constant which can be evaluated mathematically via asymptotic analysis.

Our conclusions are partially backed up using tests with Maple, MAGMA, and an XL implementation featuring Block Wiedemann as the sparse-matrix solver.

Keywords: sparse solver, Gröbner basis, XL, MQ, asymptotic analysis, F_4, F_5

1 Introduction

$\mathcal{M}Q$ (Multivariate Quadratic), or finding variables $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in (\mathbb{F}_q)^n$ from quadratic equations $p_1(\mathbf{x}) = p_2(\mathbf{x}) = \cdots = p_m(\mathbf{x}) = 0$, is an important hard problem. Instances of $\mathcal{M}Q$ appear in cryptographic situations such as a key step of many attacks known collectively as algebraic cryptanalysis.

J.-C. Faugère's F_4/F_5 algorithms are excellent system-solving algorithms both for $\mathcal{M}Q$ and for even more generalized problems with higher-degree polynomials with general applicability — that is, they work well for a large variety of systems including random ones — and are recognized as the de facto standard in the crypto community. Good commercially available implementations of generic F_5 being still sadly lacking, the F_4 implementation in MAGMA [21] is the usual yardstick against which equation-solving is measured [18].

If we limit ourselves to a somewhat theoretic context, cryptographers would like to find the best estimate of complexity of solving a random $\mathcal{M}Q$ when $m/n = c + o(1)$. It is generally believed [4] that the probability of any sub-exponential algorithm solving such random systems becomes negligible as the pameters increase. Such an algorithm would be the most important generic attack for what is known as multivariate quadratic PKCs (cf. [5]), and (iii) it determines the security of several provably secure constructions such as QUAD [4].

1.1 Questions

Despite a near monopoly of Faugère's F_4/F_5 algorithms in the crypto community, other algorithms has been proposed over F_4/F_5 in various contexts:

1. it has been noted that SAT solvers excel in specific cases;
2. other variants of Gröbner Basis methods, such as MutantXL [22,23] or GGV [17], have been claimed to have better general performance; and
3. it has was suggested that the complexity of $\mathcal{M}Q$ might be better estimated via XL with sparse-matrix solvers when $m/n = c + o(1)$.

While the superiority of "Sparse XL" variants has been suggested since 2004 [28,30], the issue of their merit has never been comprehensively settled.

In determining what circumstances favor Sparse XL over F_4/F_5, and vice versa, it is clear that systems which have a smaller difference between operating degrees of XL and F_4/F_5— smaller fields, generic ("semi-regular") systems, and more rather than less overdetermined — are better for XL. By late 2008, system-solving experts understood that if this difference is small, then XL with Sparse matrices will dominate F_4/F_5 to hold as the problem sizes get larger for generic $\mathcal{M}Q$ instances with n $m/n = c + o(1)$, provided that that certain heuristic conditions continue to hold.

With the Second Post-Quantum Cryptography Workshop at the University of Cincinnati (Oct. 17-19, 2008, Cincinnati Ohio, USA), a meeting of the Steering Committee of the workshop series was held during which Johannes Buchmann named some key research questions in PQCrypto, one being *whether the difference between the operating degrees of XL and* $\mathrm{F}_4/\mathrm{F}_5$ *approaches can be accurately bounded for random systems.*

1.2 Results

We are able to show, using saddle point asymptotic analysis that

1. in many cases of cryptographic interest, the difference in the degrees of operation for XL and F_4/F_5 is bounded tightly by a constant;
2. as n increases, the expected value of the difference approach a constant which can be rigorously determined;

3. the difference is at most 1 for many types of generic systems with $m/n = c + o(1)$, which means that the degrees of operation are usually the same for XL and F_4/F_5;
4. a specific case where the difference is unbounded is the large-field case with $f = m - n$ =constant, which is mathematically expected and explainable.

Example: For most generic large-field systems with $m/n = 2$, the degree of operation for XL and F_4/F_5 are equal — actually about 80% of the time — and in the remaining cases the difference is 1.

Example: For direct attacks on QUAD-like ciphers (where provable security reduces to an $m/n = 2$ generic $\mathcal{M}Q$), the degree of operation for XL and F_4/F_5 also differs by 1 about half of the time, and are equal the rest of the time.

1.3 Prior and Related Work

Matrix Techniques in Gröbner-Basis Computations: Most modern system-solvers compute Gröbner Bases. In the 1965 original Buchberger algorithm [8], we take equations two at a time and eliminate around their lead terms by some ordering strategy. Lazard [19] noted that since each successive step involves linear combinations of the original equations, we save work by computing and storing a batch of monomial-equation products. Further, by making each equation a row in a matrix, we enable the use of efficient and well-studied elimination algorithms in linear algebra. This is the initial appearance of the algorithm now known as XL, and leaves open the use of sparse matrix algorithms.

Initial Appearances of XL with Sparse Solver: In 2004, Yang et al mentioned the possibility that despite a higher operating degree, XL with a sparse matrix solver would be better than F_4/F_5 with a conventional solver (such as Strassen with $\omega \approx 2.8$) and can when $q = 2$ it can potentially outdo a brute force search when $m = n$ [28]. Later that year, it was noted that in by adding the "F" ("fix", or guessing) approach [29], FXL with a sparse solver would be the method with the best time complexity and discusses how to compute the optimal number of guesses. In 2006 we see an initial implementation of such an algorithm in [31], now using standard Wiedemann as the solver.

Actual Use of Sparse XL for Cryptanalysis: 2006 marks the initial cryptanalytic paper [30] where the sparsity of the matrices plays a role. In this work it was noted that using Wiedemann allows an attack to be carried out with a practical computer in Sparse XL but not in MAGMA [21] of the time, due to the smaller memory footprint.

Parallelization of Sparse XL: In [15], the authors use a tailored XL algorithm with a parallelized standard (not block) Wiedemann using a large computer and OpenMP in defeating a Rainbow/TTS scheme with a suboptimal structure.

In [24], the authors implement an XL algorithm using block Wiedemann for 32 equations and 32 variables using just 8GB of main memory (this runs out of memory in MAGMA [21] at the time, and as late as 2012).

In [9] we find a block Wiedemann optimized for XL for a variety of different fields, including $\mathbb{F}_{16}$, $\mathbb{F}_2$, and $\mathbb{F}_{31}$, using both contiguous and MPI pragmas.

Legitimatization of Sparse XL: [3] introduces an algorithm termed BooleanSolve but is effectively the same as XL using a sparse solver and guessing (fixing). The formula for evaluating monomials is also one that would be used for FXL, not one corresponding to "the Hybrid Approach" [6] which advocates guessing with F$_4$/F$_5$ instead of XL.

[3] after nearly a decade of denial of neglect against XL still does a poor job of describing prior art, but it effectively vindicates Sparse XL and affirms the asymptotic superiority of Sparse XL than F$_4$/F$_5$ for generic systems. While we cannot pretend to read minds one practical reason for this late concession is precisely the fact pointed out in this work, i.e., that the degree of operation for XL and F$_4$/F$_5$ are often the same and has a bounded difference in most random cases.

1.4 Future Work

There are several issues identified by our study.

- Complexities previously evaluated with F$_4$/F$_5$ may need to be recomputed with Sparse XL variants. [3] does this to some extent but is incomplete in this aspect.
- An obvious improvement to Sparse XL for many situations would be Sparse F$_4$ (XL2). However, an straightforward implementation of that approach would be wasteful in that it throws away previously performed work with each raised degree. A better combination between a Wiedemann-like solver and F$_4$ would instantly lead to great advances.
- The number of columns that actually appear in the final F$_4$/F$_5$ matrix, a parameter that would determine a cut-off size when XL with sparse matrices catches up to F$_4$/F$_5$, is still yet to be determined conclusively.

2 History and the Status Quo of XL vs. F$_4$/F$_5$

Notations: We will denote by $\mathbf{x}^\mathbf{b}$ the monomial $x_1^{b_1}x_2^{b^2}\cdots x_n^{b_n}$, and its degree by $|\mathbf{b}| = \sum_{i=1}^n b_i$. We will choose a degree of operation D, and let $\mathcal{T} = \mathcal{T}^{(D)} = \{\mathbf{x}^\mathbf{b} : |\mathbf{b}| \le D\}$ be the set of degree-D-or-lower monomials. Multiply each equation p_i by all monomials $\mathbf{x}^\mathbf{b} \in \mathcal{T}^{(D-\deg p_i)}$ to form the set of relations $\mathcal{R} = \mathcal{R}^{(D)} = \{\mathbf{x}^\mathbf{b} p_j(\mathbf{x}) = 0 : 1 \le j \le m,\ |\mathbf{b}| \le D - \deg p_i\}$ at degree $\le D$. $T := |\mathcal{T}^{(D)}|$ is the number of terms, and we will use the combinatorial notation $[t^k]s(t)$ for the coefficient of t^k in the Maclaurin series expansion of the function $s(t)$ in t, so $T = [t^D]\left((1-t^q)^n/(1-t)^{n+1}\right)$. We denote also by ω the exponent in the complexity of matrix multiplication/inversion. The infimum of this complexity exponent has recently been shown to be as low as 2.3727 [26], but for practical purposes is likely to be $\log_2 7 \approx 2.8$.

Basic XL: Solve $\mathcal{R}^{(D)}$ as a linear system in monomials $\mathbf{x}^\mathbf{b} \in \mathcal{T}^{(D)}$, with complexity $\propto \left(T^{(D)}\right)^\omega$.

The original XL article [12] mentioned the possibility of the Macaulay matrix reducing to a univariate equation. This happens when $I = \operatorname{span}\mathcal{R} \ge T - D$, then brute force or Berlekamp's algorithm will find the solution. However, in the overdetermined case we usually either see that $T - I = 0$ with a self-contradictory system or 1 with exactly 1 solution. Indeed, XL', or reducing to r equations in r variables when $1 < T - I < \binom{r+D}{D}$, is not known to makes a difference in any practical case known [27,29].

XL with Sparse Matrices: The Macaulay matrix $\mathcal{M}^{(D)}$ has total weight $\sim Rn^2/2$, where $R = |\mathcal{R}^{(D)}| = m|\mathcal{T}^{(D-2)}|$ is the number of equations. Hence, the linear system may be solved via (Block) Wiedemann or some similar sparse matrix solver [11,25] in $\sim \frac{3}{2}RTn^2$ multiplications. A heuristic variant [30] discards rows randomly to come down to only T rows and then solve using (Block) Wiedemann, using only $\approx \frac{3}{2}T^2n^2$ multiplications. [30] notes that this produces a single solution for most "random" overdetermined systems. If the nullity $\ell > 1$, then perhaps we dropped an essential equation, or if the system started with more than one solution. Here we must check below at every vectors of a subspace with an entry of 1 in the slot correspond to the monomial 1 ("normalized"), about $q^{\ell-1}$ points.

Why would randomly tossing rows work? Heuristically, N random vectors in $(\mathbb{F}_q)^N$ span the entire space with non-zero probability $\approx 1 - \frac{1}{q-1}$ even as $N \nearrow \infty$. Empirically, in many runs failures are even fewer and farther in between, and most singular matrices are further only of small nullity (1 or 2).

XL2 (a.k.a. MutantXL [14]): Consider starting XL at degree D and performing some kind of elimination on the equations $\mathcal{R}^{(D)}$ to attempt eliminating all the highest-degree monomials. If we fail, then raise the degree by multiplying each remaining row by every variable and repeat; if we succeed, then we have found lower-degree equations ("mutants") which can be multiplied by monomials to form new equations without raising the operating degree. In this case, with more elimination and degree-raising stages, we will usually continue to termination [27] without the degree increasing again.

F$_4$/F$_5$*:* It suffices to know that these are "better" versions of XL2 where the matrix-building and row-operation sequences are run according to certain rules to avoid redundancy, but F$_4$/F$_5$/XL2 all run at the same degree [27], which is D_{reg} for semi-regular systems. The time complexity will be bound by $\left(T^{(=D)}\right)^{\omega}$, where $T^{(=D)}$ is the number of degree-D terms, so $\binom{n}{D}$ for $q = 2$, $\binom{n+D}{D}$ for large q.

How Does Sparsity Matter? If the dimension of the matrix does not differ by a large factor, then eventually the log-complexity of XL (with Sparse Matrix Solvers) would be $2/\omega$ that of F$_4$/F$_5$, *but only if the latter cannot work with very sparse matrices*. The main determining factor for the dimension of the matrix would then be the degree of operation, which is the subject of this study.

2.1 Degrees of Operation

Small Fields Even though the claimed "proof" of the formula is mistaken, most experts expect XL to operate at the degree indicated by the heuristic formula [28, Theorem 2]:

$$D_0 := \min\{d : [t^d] \ \left((1-t)^{-n-1} \ (1-t^q)^n \ (1-t^2)^m \ (1-t^{2q})^{-m}\right) \leq 0\}.$$

Analogously, we expect F$_4$/F$_5$ to operate at what is known as “degree of regularity” (cf. [1] for $q = 2$):

$$D_{\text{reg}} := \min\{d : [t^d]\ \left((1-t)^{-n}\ (1-t^q)^n\ (1-t^2)^m\ (1-t^{2q})^{-m}\right) < 0\}.$$

Large Fields F$_4$/F$_5$/XL2 and XL operate [13,28] at (respectively)

$$D_{\text{reg}} := \min\{d : [t^d]\left((1-t)^{m-n}(1+t)^m\right) < 0\},$$

and

$$D_0 := \min\{d : [t^d]\left((1-t)^{m-n-1}(1+t)^m\right) < 0\}.$$

The former is assumed to be true from the definition of semiregularity; the latter is proved assuming the Maximal Rank Conjecture.

As functions of n and m, it is obvious that almost always $D_0(n, m) = D_{\text{reg}}(n + 1, m)$. Exceptions occur when $[t^{D_0}]\big(((1-t)^{m-n-1}(1+t)^m\big) = 0$, the most common case being $m - n = 2$ and n odd.

Operating Degree as a Root of a Function via Integrals: The degree d coefficient of the Maclaurin series of $f(t)$ is given by a contour integral $S_f(d) := (2\pi i)^{-1} \oint \left(f(z)\ z^{-(d+1)}\right) dz$. The power of the first nonpositive (or resp. negative) coefficient of $f(t)$ is then the smallest integer no less (resp. greater) than the smallest positive real root of S_f. Here we will denote by $\widehat{D_0}$ and $\widehat{D_{\text{reg}}}$ the smallest positive roots of the corresponding integral functions, hence $D_0 = \lceil \widehat{D_0} \rceil$ and $D_{\text{reg}} = \lfloor \widehat{D_{\text{reg}}} \rfloor + 1$.

Known Asymptotic Results [1, 2, 29]: For $m = (c + o(1))n$ where c is a constant and for any $\mathbb{F}_q$, we have D_0 and D_{reg} also equal to $(w + o(1))n$, where w depends only on c and q. $\lg T$ will also asymptotically proportional to n. For any q and ω (i.e., algorithm of elimination), one guesses up to a $c \sim m/n$ to optimize the number of field multiplications one makes.

2.2 A Note on Why Is not XL Better?

[27,29] advocated XL with Sparse solvers as asymptotically better than F$_4$/F$_5$ in the generic case, especially as the memory size gets larger. Indeed, one might imagine that F$_4$/F$_5$ is no match for XL with sparse matrices if the former works with degree 3 or $\log_2 7$ complexity in matrix size, and the latter degree $2 + o(1)$. Yet, the question is, if XL is fundamentally better, why is there not such a report?

Linear Algebra Implementation: It is understandable that the per-multiplication cost in a sparse matrix solver is larger than that in solving a dense linear system, because linear algebra with dense systems is a well-known subject. Even linear algebra in finite fields are optimized very well using tricks like [20] (often known erroneously as the Method of 4 Russians, due to Lupanov-Kronrod).

The Choice of Parameters However, the difference in speed caused by implementation issues is usually a constant or at most polynomial factor. We will show later that the decisive factor was a different one: Typically such investigations examined a large-field, $m - n =$constant case, which as we will see below is exactly the worse case for XL among generic systems with $m/n = c + o(1)$.

2.3 Matrix Operations in XL vs. F$_4$/F$_5$ if both $\frac{D_0}{n}$ and $\frac{D_{\text{reg}}}{n} = w + o(1)$

Difference in Operating Degree Means Difference in Size: If $D_0 > D_{\text{reg}}$, then the XL matrix dimension increases with respect to F$_4$/F$_5$/XL2 by an extra factor of

$$\frac{(n + D_{\text{reg}})(n + D_{\text{reg}} + 1) \cdots (n + D_0 - 1)}{(D_{\text{reg}} + 1)(D_{\text{reg}} + 2) \cdots D_0} \approx \left(1 + \frac{1}{w}\right)^{D_0 - D_{\text{reg}}}.$$

We could say each increment of $D_0 - D_{\text{reg}}$ costs XL (versus F$_4$) a factor of $\geq 3\times$.

Other Factors: The Sparsity and the structure of the extended Macaulay matrix works in favor of XL with sparse solvers in terms of better memory footprint and lower complexity. Everything else will be operating against XL. Here we list some differences of XL vs. F$_4$/F$_5$.

Memory Use and Storage for Matrix: In F$_4$/F$_5$, recent lectures (ECC 2011 [16] and earlier, Polynomial Equations Solving workshop at KTH, Stockholm, Sweden) gave density of the non-zero entries involved in the matrix steps as about $d^{-1}\sqrt{(6/\pi n)}$ for random systems, where d is the degree of the polynomials, so the matrix is fairly dense, with total weight $\sim T^2/\text{poly}(n)$.

As for the XL Sparse version, each row in the Extended Macaulay Matrix has essentially the same number of entries, which is $\sim n^2/2$. There are various ways to compress a sparse system. To take it to the extremes, one could simply store the original equations and generate all of the Macaulay matrix on the fly. In practice, one need to store column indices in each block of rows to avoid recomputation. The total Macaulay matrix storage is thus $Tn\ \text{polylog}(n)$.

However, there is one operational detail which sometimes offsets some of the advantages of XL which is that when parallelizing, memory needs to be handled in cache lines and each core needs full vectors data each load in any parallelized solver like Block Wiedemann, and storage is needed for source and destination. So the memory footprint for Block Wiedemann vectors is $2T\nu s_v$ where ν is the number of cores and s_v is the vector length to fit cache lines. For small-to-medium cases this is often larger than the matrix size.

Extra Columns: The number of terms in XL is larger by a factor of $T^{(D)}/T^{(=D)} = \binom{n+D}{D}/\binom{n+D-1}{D} = \frac{n+D}{n} \sim (1 + w + o(1))$ even if the two degrees are equal – but the constant is not far removed from 1.

A more important issue: some terms may be completely eliminated (and with it the associated columns and pivots) during F$_4$/F$_5$ and never appear again. To give an example, there are 2^{21} monomials of degree 9 where F_4/F_5 solves $\mathcal{M}Q$ with $(n, m) = (17, 19)$, but the MAGMA output indicates that the matrix is only about 2^{34} bytes, which indicates that the matrix is much emptier or smaller than the 2^{42} bytes that a raw extrapolation would indicate even taking into account the sparsity estimate given above. This phenomenon was also described in [22, 23] ("the partial enlargement strategy"), and noted in passing by F_4/F_5 investigators.

Extraneous Rows: In F_4, there are some extraneous rows; in F_5, there are no extraneous rows (that will reduce to zero) generated at the cost of some restrictions on linear algebra; in XL, there are *many* extra rows in the Macaulay matrix, but any random row-tossing scheme would cut it down to a square matrix. We thus expect the matrix dimensions to be relatively close at the same degree.

How and When XL might be Better than F_4/F_5 We expect that the ratio of the per-multiplication cost in the linear algebra is going to be more or less constant with good programming; it is also something that is harder for us to control. However, it is easier to find cases that favor XL over F_4/F_5 if we choose cases where there *is not* a large difference in the operating degrees.

The attacks in [30] dealt with random $\mathcal{M}Q$ where $m = 2n$, which is related to the provable security of QUAD, is one such example. The XL-with-Block-Wiedemann implementation of [9] on a 32-core Xeon E5620 2.4GHz mini-cluster, takes 577 seconds with $(n, m) = (24, 48)$ over $\mathbb{F}_{16}$. MAGMA-2.17 on a Xeon X7550 2.0GHz takes 68628 seconds when $(n, m) = (24, 48)$. This is a good case for XL — the operating degrees are the same; if we change to the parameters $(n, m) = (23, 46)$, XL operates at a higher degree (6 vs 5). This is consistent with the above impression that XL does better where the difference $D_0 - D_{\mathrm{reg}}$ is small, in particular zero. In the remainder of this article, we try to find out which parameters tend to satisfy this property.

3 Degrees of Operation for F_4/F_5 vs. XL and Asymptotics

In this section, we examine the difference in the operating degree of F_4/F_5 vs XL. Clearly $\lceil \widehat{D_0} - \widehat{D_{\mathrm{reg}}} \rceil \geq D_0 - D_{\mathrm{reg}} \geq \lfloor \widehat{D_0} - \widehat{D_{\mathrm{reg}}} \rfloor \geq 0$. We show that $\widehat{D_0} - \widehat{D_{\mathrm{reg}}}$ is asymptotically large in large-field, almost-square systems only. We then discuss how this reflects on the practical complexities of XL vs F_4/F_5. All the details would be included in a future full version.

3.1 Large Fields ($q > D$), Barely Overdetermined ($m/n = 1 + o(1)$) Cases

We observe empirically that $D_0 - D_{\mathrm{reg}}$ is seldom zero. If $m, n \to \infty$ while $m - n = f > 1$ fixed, then the degree of regularity D_{reg} for a system of m quadratic equations in n variables is asymptotically given by $\widehat{D_{\mathrm{reg}}} = \frac{m}{2} - h_{f,1} \cdot \sqrt{\frac{m}{2}} \cdot (1 + o(1))$ [2], where $h_{f,1} = \sqrt{2f + 1} + O(f^{-1/6})$ is the largest zero of the Hermite polynomial of order f. Hence $\widehat{D_0} - \widehat{D_{\mathrm{reg}}} = (h_{f,1} - h_{f-1,1})\sqrt{\frac{m}{2}}\,(1 + o(1))$.

Practical Implication for XL vs. F$_4$/F$_5$*:* It is difficult for XL with Sparse solvers to catch up to F$_4$/F$_5$, because $\widehat{D_0} - \widehat{D_{\text{reg}}} \geq 1$, and at some point (which for $m - n = 2$ is $(20, 22)$) becomes always 2 or more, which means that the number of monomials in XL is $> 10\times$ or more that of F$_4$/F$_5$, without taking into account the pivots that disappear from the later stages of F$_4$/F$_5$. In fact, the only practical way that XL would be better might be because the matrix would be too large to handle in F$_4$/F$_5$.

3.2 Large Fields ($q > D$), QUAD-like ($m/n \sim \alpha > 1$) case:

The asymptotic expansion for $\widehat{D_{\text{reg}}}$ for large q is given by [2]:

$$\widehat{D_{\text{reg}}} = (\alpha - \frac{1}{2} - \sqrt{\alpha(\alpha-1)})n + \frac{-a_1}{2(\alpha(\alpha-1))^{\frac{1}{6}}} n^{\frac{1}{3}} - \left(2 - \frac{2\alpha - 1}{4(\alpha(\alpha-1))^{\frac{1}{2}}}\right) + O(\frac{1}{n^{1/3}}).$$

So in a typical case for QUAD, $m = 2n$ ($\alpha = 2$), and

$$\widehat{D_{\text{reg}}}(n, 2n) \approx 0.0858n + 1.0415n^{1/3} - 1.4697 + O(n^{-1/3}).$$

It is worth noting that in asymptotic analysis of the root of a function using coalescent saddle points [10], due to the characteristics of the Airy integral expansions, typically the expansion is a series in $n^{-1/3}$ missing the second term, or $f(n) := a_0 n^\beta + a_2 n^{\beta - 2/3} + a_3 n^{\beta-1} + a_4 n^{\beta - 4/3} + \cdots$, so, on first thought we would expect a difference in the next-to-leading term, or $\widehat{D_0}(n, 2n) - \widehat{D_{\text{reg}}}(n, 2n) = a_2 n^{1/3} + O(1) \nearrow \infty$ as $n \to \infty$, *except that it doesn't but rather approaches a constant near* 0.207.

Indeed, we can carry through the same Coalescent Saddles computations to find that

$$\widehat{D_0}(n, 2n) \approx 0.0858n + 1.0415n^{1/3} - 1.2626 + o(1), \tag{1}$$

$$\text{or} \approx \quad \widehat{D_{\text{reg}}}(n, 2n) + 0.2071 + o(1). \tag{2}$$

We verified the 0.207 asymptotic for semiregular systems over the range $n = 10 \cdots 120000$. Practically, this number starts at around 1/4 in the practical range and decreases toward 0.207. The upshot is that **when $m/n \approx 2$, more than three quarters of the time XL and XL2/F$_4$/F$_5$ runs at the same degree.**

Practical Implications for XL vs. F$_4$/F$_5$*:* $D_0(n, 2n) - D_{\text{reg}}(n, 2n) \leq 1$ for almost all n. Furthermore, if we regard the linear and $n^{1/3}$ terms as supplying a random fractional part between 0 and 1, *we can expect that exactly* 20.7% *among all n have operating degrees $D_0(n, 2n)$ and $D_{\text{reg}}(n, 2n)$ differ by* 1. This puts XL in a (relatively speaking) good position compared to F$_4$/F$_5$. Degree increases (or drops) in XL/F$_4$/F$_5$ matter a lot because the number of monomials increases by a factor that is often between $4\times$ to $6\times$ but asymptotically a factor of $\approx (1 + 0.0858)/(0.0858) \approx 12.66$.

If we fix q and increase n, the system ceases to be "large-field" in that eventually $D_{\text{reg}} > q$. However, for practical attacks we expect the degree drop $D_0 - D_{\text{reg}}$ to be limited to 1 (see following sections). Empirically size of the matrix in the MAGMA F$_4$ is also somehow larger for the same T for $m/n = 2$ cases than for $m/n \sim 1$ cases. This is again understandable heuristically in the sense that n is larger but D is smaller if we compare an $m/n = 2$ instance with an equal-T instance where $m/n \approx 1$, which means far fewer eliminated columns and pivots.

We conclude that in QUAD ($m/n = 2$) type instances, estimation of cryptographic complexities must take into account attacks using XL as opposed to $\mathrm{F}_4/\mathrm{F}_5$, *if not using the former outright.*

An Explanation of 0.207*:* Why does it happen here that the $n^{1/3}$ term coefficient does not change? There is a good reason for that. A heuristic "proof" is that we can write the uniform asymptotic expansion as follows

$$\widehat{D_{\text{reg}}} = \left(1 - \frac{\alpha^{-1}}{2} - \sqrt{(1-\alpha^{-1})}\right) m + \frac{-a_1 \cdot (\alpha^{-1})^{2/3}}{2(1-\alpha^{-1}))^{\frac{1}{6}}} m^{\frac{1}{3}} - \left(2 - \frac{2-\alpha^{-1}}{4(1-\alpha^{-1})^{\frac{1}{2}}}\right) + O(\frac{1}{m^{\frac{1}{3}}}).$$

Hence, if we write $\widehat{D_{\text{reg}}}(n, \alpha n) = f(\alpha^{-1}, m)$, then

$$\begin{aligned}\widehat{D_0}(n,2n) - \widehat{D_{\text{reg}}}(n,2n) &= f\left(\left(\frac{1}{2} + \frac{1}{2n}\right), 2n\right) - f\left(\frac{1}{2}, 2n\right) \\ &\approx \frac{1}{2n} \cdot \left.\frac{\partial f}{\partial(\alpha^{-1})}\right|_{\alpha^{-1}=\frac{1}{2}, m=2n} = \left(\frac{\sqrt{2}-1}{2}\right) + o(1),\end{aligned}$$

which explains why $D_0(n, 2n+k) - D_{\text{reg}}(n, 2n+k)$ is also on average 0.207 as soon as n gets somewhat large. We verified this for integers from $k = -10, \dots, 10$. Similarly, we can verify that for $m/n \approx 1.5$ and 2.5, the degree drop converges to 0.367 and 0.145, respectively. This analysis can be made rigorous (and similarly for the heuristic "proof" below) with some complex analysis.

The Distinctiveness of the $m/n = 1 + o(1)$ *case:* The reason that the Large Fields ($q > D$), Barely Overdetermined case is so different is that $\alpha = 1$ is a singularity and hence there is no way to take a differentiative at that point with respect to α.

3.3 Small-Field Cases

For $\mathbb{F}_2$, $\mathbb{F}_3$, and $\mathbb{F}_4$ and all $\alpha = m/n > 1$, $D_0 - D_{\text{reg}} \leq 1$ for all practical cases.

$\mathbb{F}_2, m = n$ case: this resembles the large-field, $m/n = 2$ case in behavior. In part this is because we can think of the "field equations" $x_i^2 = x_i$ as n more equations. Note that as the smallest field, the behavior of $\mathbb{F}_2$ is not truly representative of all small fields, but $\mathbb{F}_2$ is so important in cryptography we simply have to use it as the example. If we carry out the requisite coalescent saddle points computations, we find that

$$\widehat{D_0} - \widehat{D_{\text{reg}}} = 0.2339 + o(1).$$

Just like for large q cases, this implies that (for all practical purposes) $D_0 - D_{\text{reg}} \leq 1$ and is only non-vanishing on less than a quarter of possible n's on average. We verified that this average is roughly correct by using Maple to compute the series up to $n = 10000$.

Practical Implications: In the practical range, we expect that Sparse XL will do better than F_4/F_5 as is first claimed in [28], verified by [24] and conceded in [3]. *However, this is not a good case for Sparse XL because XL will be comfortably outrun by brute force searches.* In the parallelized Block Wiedemann implementations of XL of [9], it was seen that for 35 variables in 35 equations takes 45571s on a test machine with 64 2.3GHz AMD Bulldozer cores and 256GB of contiguous main memory. The same machine would use $<$ 1s on a brute-force search [7].

$\mathbb{F}_2$, $m = 2n$ case: Similar to the previous case, we did Coalescent Saddle Point analysis to find that

$$\widehat{D_0} - \widehat{D_{\text{reg}}} = 0.1169 + o(1).$$

This implies that $D_0 - D_{\text{reg}} \leq 1$ in practice for all D_0, and the proportion of n where $D_0 - D_{\text{reg}} = 1$ is a scant 15% within or less for $n \leq 10000$.

This is a more interesting case to be talking about XL because of two reasons. One is that this is the security assumption for QUAD stream ciphers and variants. The other is that it is easier for XL to beat brute force. Unfortunately, it is not *that* easy. For an example, at $n = 96$, $m = 192$, Sparse XL is projected to take $2^{94.6}$ field multiplications. Each field multiplication takes about 1/20 of a cycle. But if we consider the memory footprint, XL still loses badly to a brute-force attack. If we take memory size into account, to use XL in $\mathbb{F}_2$, we should continue to guess until m/n is close to 3.

Other Small Fields We can verify that for $\mathbb{F}_3$ and $\mathbb{F}_4$ n-variables-n-equation systems have $D_0 - D_{\text{reg}} \leq 1$, just like $\mathbb{F}_2$. In fact,

$$\begin{aligned}
\widehat{D_0}(n,n;3) - \widehat{D_{\text{reg}}}(n,n;3) &= 0.3660 + o(1)\\
\widehat{D_0}(n,2n;3) - \widehat{D_{\text{reg}}}(n,2n;3) &= 0.1650 + o(1)\\
\widehat{D_0}(n,n;4) - \widehat{D_{\text{reg}}}(n,n;4) &= 0.4940 + o(1)\\
\widehat{D_0}(n,2n;4) - \widehat{D_{\text{reg}}}(n,2n;4) &= 0.1912 + o(1)
\end{aligned}$$

Using maple, one can check for $m = n$ that for roughly 38% and 53% of all $n < 5000$ that $D_0 - D_{\text{reg}} = 1$ for $\mathbb{F}_3$ and $\mathbb{F}_4$ respectively, and the other times the two degrees are equal. What this means is that generic equations in smaller fields generally favor XL over F_4/F_5. However (although a little surprising to begin with), when facing a system with $m = n$ in these small fields we need to check whether brute force is best also.

Heuristic Evaluation of $\widehat{D_0} - \widehat{D_{\text{reg}}}$ for small fields: Without entering into the complex analysis required to prove all of the above rigorously, we will compute an example the asymptotic behavior of $\widehat{D_0}(n,2n;2) - \widehat{D_{\text{reg}}}(n,2n;2)$. Here $\widehat{D_{\text{reg}}}(n,2n;2)$ and $\widehat{D_0}(n,2n;2)$ are respectively the smallest positive root of

$$S_1(d) := \frac{1}{2\pi i}\oint \frac{(1+z)^n \, dz}{z^{d+1}\,\left(1+z^2\right)^{2n}}, \quad \text{and} \quad S_2(d) := \frac{1}{2\pi}\oint \frac{(1+z)^n \, dz}{(1-z)\, z^{d+1}\,\left(1+z^2\right)^{2n}}.$$

Let $w = d/n$ and consider S_1 and S_2 as special cases of the following contour integral

$$S(n; w; \alpha, \beta, \gamma) := \oint \frac{dz}{2\pi i z} \left(\frac{(1+z)^\alpha}{z^w \, (1+z^2)^\beta \, (1-z)^\gamma} \right)^n .$$

To evaluate this we need the following equation in z to have double roots:

$$\frac{-w}{z} + \frac{\alpha}{1+z} - \frac{2\beta z}{1+z^2} + \frac{\gamma}{1-z} = 0.$$

If we let $w := F(\alpha, \beta, \gamma)$ represent the smallest positive real $w = d/n$ that allows double roots for z, then we see that $F(\alpha, \beta, 0)$ is the coefficient of the $\Theta(n)$ term in the asymptotic expansion of $\widehat{D_{\text{reg}}}(\alpha n, \beta n; 2)$ and $\widehat{D_0}(\alpha n, \beta n; 2)$, and if we skip all the analysis, we eventually get to

$$\widehat{D_0}(n, 2n; 2) - \widehat{D_{\text{reg}}}(n, 2n; 2) = \left(F(1, 2, \frac{1}{n}) - F(1, 2, 0) \right) n = \left. \frac{\partial F}{\partial \gamma} \right|_{\alpha=1, \gamma=0} + o(1), \tag{3}$$

which we may evaluate with implicit differentiation to obtain the 0.1169 above.

3.4 Direct Attacks on QUAD

In a direct attack against QUAD we face this problem: take random polynomials $\mathbf{P} = (P_1, \ldots, P_n)$ and $\mathbf{Q} = (Q_1, \ldots, Q_n)$ in the variables $\mathbf{x} = (x_1, \ldots, x_n)$. Solve ℓn equations for $\mathbf{x}$ using vectors $\mathbf{y}_1, \ldots, \mathbf{y}_\ell$:

$$\mathbf{y_1} = \mathbf{P}(\mathbf{x}), \, \mathbf{y_2} = \mathbf{P}(\mathbf{Q}(\mathbf{x})), \, \mathbf{y_3} = \mathbf{P}(\mathbf{Q}(\mathbf{Q}(\mathbf{x}))), \ldots .$$

This arises from studying the security of the stream cipher QUAD [4]. [30] suggested this direct attack in the known-plaintext setting and verified empirically that this system behaves like random systems (i.e., a system with n random quadratic equations, n random quartic equations, and so on) if one tries to solve it with Gröbner basis methods, including XL.

Our investigations and tests show that the systems created in direct algebraic attacks on QUAD-like systems also have $D_0 - D_{\text{reg}} \leq 1$ if we assume semi-regularity, and hence we expect XL to overtake F$_4$/F$_5$ *as the best estimate of complexities for moderately large n.*

$\mathbb{F}_2$ *cases:* We compare the operating degrees of XL and F$_4$/F$_5$ as given by [30, Sec. 4.5]:

$$D_0(\texttt{QUAD}(2, n, n)) = \min \left\{ D : [t^D] \, \tfrac{(1+t)^n}{(1-t)} \, \left((1+t^2)(1+t^4) \cdots (1+t^{2^\ell}) \right)^{-n} \leq 0 \right\},$$

$$D_{\text{reg}}(\texttt{QUAD}(2, n, n)) = \min \left\{ D : [t^D] \, \tfrac{(1+t)^n}{\left((1+t^2)(1+t^4) \cdots (1+t^{2^\ell})\right)^n} < 0 \right\}.$$

As $\ell \nearrow \infty$, we have $\left((1+t^2)(1+t^4) \cdots (1+t^{2^\ell}) \right) = (1-t^2)^{-1} \left(1 - t^{2^\ell} \right) \longrightarrow (1-t^2)^{-1}$, hence

$$D_0(\texttt{QUAD}(2, n, n)) = D_0(n, 2n; \text{large } q), \; D_{\text{reg}}(\texttt{QUAD}(2, n, n)) = D_{\text{reg}}(n, 2n; \text{large } q).$$

So the operating-degree difference of XL and F$_4$/F$_5$ bounded by 1 and average 0.2071.

Large-Field Cases: As found by [30, Sec. 4.4], using more than the quartics does not lead to substantial gains. We will hence restrict ourselves to the direct attack using only quadratics and quartics:

$$D_0(\texttt{QUAD}(\text{large } q, n, n)) = \min\left\{D : [t^D] \ \left((1-t)^{-(n+1)}(1-t^2)(1-t^4)\right)^n < 0\right\},$$
$$D_{\text{reg}}(\texttt{QUAD}(\text{large } q, n, n)) = \min\left\{D : [t^D] \ \left((1-t)^{-n}(1-t^2)(1-t^4)\right)^n < 0\right\}.$$

We discover that $D_0(\texttt{QUAD}(\text{large } q, n, n)) - D_{\text{reg}}(\texttt{QUAD}(\text{large } q, n, n))$ is zero for roughly half of all n and 1 for the other half. This might seem surprising but again this can be explained as follows: Let $R(\alpha)$ be the smallest positive w that gives a double root to $\frac{d}{dz}(1-z)^\alpha(1+z)^2(1+z^2)z^{-w} = 0$. Then as before $\widehat{D_{\text{reg}}}(\texttt{QUAD}(\text{large } q, n, n)) = n(R(1) + o(1))$ while $\widehat{D_0}(\texttt{QUAD}(\text{large } q, n, n)) = n(R(1 - \frac{1}{n}) + o(1))$. The implicit function theorem lets us find $R'(1)$ and derive (heuristically, but can be made rigorous):

$$\widehat{D_0}(\texttt{QUAD}(\text{large } q, n, n)) - \widehat{D_{\text{reg}}}(\texttt{QUAD}(\text{large } q, n, n)) = 0.4843 + o(1).$$

We note here that the asymptotic result is similar if we include higher-order equations.

4 Discussion and Concluding Remarks:

In this paper, we discuss the difference in the degrees of operation of the XL and the F_4/F_5 alforithms (and all similar algorithms such as MutantXL/XL2 [23], or the GGV algorithm of [17]) for multivariate systems with randomly chosen coefficients, where the ratio of the number of equations to the number of variables is nearly constant. We show that usually the difference is small. In fact, for most cryptographically relevant cases, it is at most one, and the expectation value over many possible set of parameters can be evaluated precisely using asymptotic analysis.

The inevitable conclusion is that for generic/random and mildly overdetermined systems with $m/n = c + o(1)$, XL with sparse matrices may be a better way to find roots than any of the more advanced methods. This vindicates the conjectures of [27,29] regarding XL with Sparse solvers, and is consistent with the recent article [3] which implicitly assumes a sparse matrix method and XL rather than F_4/F_5.

Future work remains to determine the best way to implement similar methods using Wiedemann type solvers. Here practical study is made difficult as so much details about F_5 and MAGMA-F_4 are unknown, but we believe that we have shed some light on the comparison of XL vs. F_4/F_5 in theory. We are preparing a full version for journal publication.

Acknowledgements

Thanks are due for their sponsorship to Academia Sinica (for an Career Advancement Award to BY) and to the National Science Council of Taiwan for grant NSC100-2628-E-001-004-MY3.

Many more thanks and appreciation are due from BY and CMC to Johannes Buchmann, who over many years had been a good friend who offers sage advice, an esteemed and supportive elders of our trade, as well as a valued colleague and co-author.

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. Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Pierre-Jean Spaenlehauer. On the complexity of solving quadratic boolean systems. *Journal of Complexity*, 29(1):53–75, 2013. ISSN 0885-064X.
4. Côme Berbain, Henri Gilbert, and Jacques Patarin. QUAD: A practical stream cipher with provable security. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 109–128. Springer, 2006.
5. Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors. *Post Quantum Cryptography*. Springer-Verlag Berlin, 1st edition, 2008. ISBN 3-540-88701-6.
6. L. Bettale, J.-C. Faugère, and L. Perret. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology*, 3(3):177–197, 2010.
7. Charles Bouillaguet, Hsieh-Chung Chen, Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, Adi Shamir, and Bo-Yin Yang. Fast exhaustive search for polynomial systems in $\mathbf{F_2}$. In Stefan Mangard and François-Xavier Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pages 203–218. Springer, 2010.
8. B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*. PhD thesis, Innsbruck, 1965.
9. Chen-Mou Cheng, Tung Chou, Ruben Niederhagen, and Bo-Yin Yang. Solving quadratic equations with xl on parallel architectures. In Emmanuel Prouff and Patrick Schaumont, editors, *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pages 356–373. Springer, 2012.
10. C. Chester, B. Friedman, and F. Ursell. An extension of the method of steepest descents. *Proceedings of Cambridge Philosophical Society*, 53:599–611, 1957.
11. Don Coppersmith. Solving homogeneous linear equations over $\mathrm{GF}(2)$ via block wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, January 1994.
12. Nicolas T. Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology — EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 392–407. Bart Preneel, ed., Springer, 2000. Extended Version: `http://www.minrank.org/xlfull.pdf`.
13. Claus 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. Jintai Ding, Johannes Buchmann, Mohamed Saied Emam Mohamed, Wael Said Abd Elmageed Mohamed, and Ralf-Philipp Weinmann. Mutant XL. talk at the First International Conference on Symbolic Computation and Cryptography (SCC 2008), Beijing, 2008.
15. Jintai Ding, Bo-Yin Yang, Chia-Hsin Owen Chen, Ming-Shing Chen, and Chen-Mou Cheng. New differential-algebraic attacks and reparametrization of rainbow. In *Applied Cryptography and Network Security*, volume 5037 of *Lecture Notes in Computer Science*, pages 242–257. Springer, 2008. cf. `http://eprint.iacr.org/2008/108`.
16. Jean-Charles Faugère. Solving efficiently structured polynomial systems and applications in cryptology. http://ecc2011.loria.fr/slides/faugere.pdf, September 2011. Talk at ECC 2011, 9:30 AM on Sep. 20, 2011.

17. Shuhong Gao, Yinhua Guan, and Frank Volny. A new incremental algorithm for computing groebner bases. In Wolfram Koepf, editor, *ISSAC*, pages 13–19. ACM, 2010.
18. Antoine Joux and Vanessa Vitse. A variant of the F4 algorithm. In Aggelos Kiayias, editor, *CT-RSA*, volume 6558 of *Lecture Notes in Computer Science*, pages 356–375. Springer, 2011.
19. Daniel Lazard. Gröbner-bases, Gaussian elimination and resolution of systems of algebraic equations. In *EUROCAL 83*, volume 162 of *Lecture Notes in Computer Science*, pages 146–156. Springer, March 1983.
20. O. B. Lupanov. On rectifier and contact-rectifier circuits. *Akademii Nauk SSSR*, 111:1171–1174, 1956. ISSN 0002ąV3264.
21. MAGMA project, Computational Algebra Group, University of Sydney. The MAGMA computational algebra system for algebra, number theory and geometry. http://magma.maths.usyd.edu.au/magma/.
22. Mohamed Saied Emam Mohamed, Daniel Cabarcas, Jintai Ding, Johannes Buchmann, and Stanislav Bulygin. MXL$_3$: An efficient algorithm for computing Gröbner bases of zero-dimensional ideals. In Donghoon Lee and Seokhie Hong, editors, *ICISC*, volume 5984 of *Lecture Notes in Computer Science*, pages 87–100. Springer, 2009.
23. Mohamed Saied Emam Mohamed, Wael Said Abd Elmageed Mohamed, Jintai Ding, and Johannes Buchmann. MXL2: Solving polynomial equations over GF(2) using an improved mutant strategy. In Johannes Buchmann and Jintai Ding, editors, *PQCrypto*, volume 5299 of *Lecture Notes in Computer Science*, pages 203–215. Springer, 2008.
24. Wael Said Abdelmageed Mohamed, Jintai Ding, Thorsten Kleinjung, Stanislav Bulygin, and Johannes Buchmann. PWXL: A parallel Wiedemann-XL algorithm for solving polynomial equations over GF(2). In Carlos Cid and Jean-Charles Faugère, editors, *Proceedings of the 2nd International Conference on Symbolic Computation and Cryptography*, pages 89–100, June 2010.
25. Douglas Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, IT-32(1):54–62, 1976.
26. Virginia Vassilevska Williams. Breaking the Coppersmith-Winograd barrier. `www.cs.berkeley.edu/~virgi/matrixmult.pdf`, 2011.
27. Bo-Yin Yang and Jiun-Ming 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.
28. Bo-Yin Yang and Jiun-Ming Chen. Theoretical analysis of XL over small fields. In *ACISP 2004*, volume 3108 of *Lecture Notes in Computer Science*, pages 277–288. Springer, 2004.
29. Bo-Yin Yang, Jiun-Ming Chen, and Nicolas Courtois. On asymptotic security estimates in XL and Gröbner bases-related algebraic cryptanalysis. In *ICICS 2004*, volume 3269 of *Lecture Notes in Computer Science*, pages 401–413. Springer, Oct. 2004.
30. Bo-Yin Yang, Owen Chia-Hsin Chen, Daniel J. Bernstein, and Jiun-Ming Chen. Analysis of `QUAD`. In Alex Biryukov, editor, *FSE*, volume 4593 of *Lecture Notes in Computer Science*, pages 290–307. Springer, 2007.
31. Bo-Yin Yang, Owen Chia-Hsin Chen, and Jiun-Ming Chen. The limit of XL implemented with sparse matrices. Workshop record, PQCrypto workshop, Leuven 2006. http://postquantum.cr.yp.to/pqcrypto2006record.pdf.