# All in the XL Family: Theory and Practice

Bo-Yin Yang[1,⋆] and Jiun-Ming Chen[2]

[1] Department of Mathematics, Tamkang University, Tamsui, Taiwan
by@moscito.org
[2] Chinese Data Security, Inc., & National Taiwan U
jmchen@math.ntu.edu.tw

**Abstract.** The XL (eXtended Linearization) equation-solving algorithm belongs to the same extended family as the advanced Gröbner Bases methods $\mathbf{F_4}/\mathbf{F_5}$. XL and its relatives may be used as direct attacks against multivariate Public-Key Cryptosystems and as final stages for many "algebraic cryptanalysis" used today. We analyze the applicability and performance of XL and its relatives, particularly for generic systems of equations over medium-sized finite fields.

In examining the extended family of Gröbner Bases and XL from theoretical, empirical and practical viewpoints, we add to the general understanding of equation-solving. Moreover, we give rigorous conditions for the successful termination of XL, Gröbner Bases methods and relatives. Thus we have a better grasp of how such algebraic attacks should be applied. We also compute revised security estimates for multivariate cryptosystems. For example, the schemes SFLASH$^{v2}$ and HFE Challenge 2 are shown to be unbroken by XL variants.

**Keywords:** algebraic analysis, finite field, Gröbner Bases, multivariate quadratics, multivariate cryptography, XL.

## 1 Introduction

Public Key Cryptography depends on the intractibility of "hard problems". Solving a system of quadratic equations over a finite field is one such (known to be NP-hard, [33]) problem. Further, often in a cryptographical primitive we find a polynomial system of equations to hold with good probability. This is called *algebraic cryptanalysis*, currently a very hot topic. Ergo, knowing how fast we can solve polynomial systems is important.

XL is an equation-solving method related to Gröbner Bases ([2, 54]). It was proposed[1] by Courtois-Klimov-Patarin-Shamir ([20]). Claims of cryptanalysis involving XL-like system-solving have been made against many primitives: stream

⋆ Supported by National Science Council of Taiwan under grant NSC 93-2115-M-032-008.

[1] XL is often regarded as a descendant of Kipnis-Shamir's relinearization ([37]), used in an algebraic attack on HFE, but we will discuss only XL-related methods from now on.

C. Park and S. Chee (Eds.): ICISC 2004, LNCS 3506, pp. 67–86, 2005.

ciphers like Toyocrypt ([15]) and $E0$ (the Bluetooth protocol, [16]), block ciphers like Rijndael/AES and Serpent ([21]), and multivariate PKC's like HFE and SFLASH$^{v2}$ ([17]).

XL does not operate on underdetermined systems, we must first take guesses to make it determined or over-determined. *Henceforth we concern ourselves with solving the system* $\ell_1(\mathbf{x}) = \ell_2(\mathbf{x}) = \cdots = \ell_m(\mathbf{x}) = 0$ *of* $m \geq n$ *(quadratic unless otherwise specified) equations in* $n$ *variables* $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ *over a field* $K = \mathrm{GF}(q)$.

We will study the time complexity of XL- and Gröbner-Bases-related algorithms. For generic systems, this depend primarily on the minimum degree of operation, which varies with $m$ and $n$ and other parameters. We hope to achieve the following:

- obtain exact and asymptotic time complexity of several XL-like methods; and hence:
- show some previous claims of cryptanalysis to be over-optimistic, and give updated security estimates for the primitives of SFLASH$^{v2}$ and HFE challenge 2 (neither of which now decreasing below $2^{80}$) by various methods;
- demonstrate that XL with the XL2 adjunct is a primitive version of $\mathbf{F_5}$.

## 2 The XL Algorithm

The "Basic XL" ([24] terms it "reduced XL") at degree $D$ proceeds as follows:

1. "X" is for eXtend (or multiply). Generate equations $\mathcal{R}^{(D)} = \{\mathbf{x}^{\mathbf{b}}\ell_i(\mathbf{x}) = 0 : i = 1 \cdots m, |\mathbf{b}| \leq D - 2\}$. $|\mathbf{b}| = \sum_i b_i$ is the degree of monomial $\mathbf{x}^{\mathbf{b}} = x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$.
2. "L" is for Linearize. Run an elimination on the equations $\mathcal{R}^{(D)}$, treating each monomial $\mathbf{x}^{\mathbf{b}}$ in the set $\mathcal{T} = \mathcal{T}^{(D)}$ of monomials of total degree $\leq D$ as a variable. The number of variables and equations are denoted $T$ and $R$ respectively. The number of independent equations (i.e., the rank of the system, denoted $I$) cannot exceed $T-1$ if the original system has a solution. Indeed, if $I = T - 1$ we expect the algorithm to terminate with a unique solution. However, it is sufficient that the elimination results in an equation to solve for (say) $x_1$.
3. *If necessary, solve the univariate equation giving* $x_1$*, and repeat as needed.*

If solving $M$ linear equations in $N$ variables takes $E(N, M)$, then XL runs in time

$$C_{\mathrm{XL}} = E(T, R) = E\left(\tbinom{n+D}{D}, m\tbinom{n+D-2}{D-2}\right), \tag{1}$$

for larger fields because $R = m\binom{n+D-2}{D-2}$ and $T = \binom{n+D}{D}$. If we are dealing with small fields, then both $T$ and $R$ would be smaller. A reasonable terminating condition is then $I \geq T - \min(D, q-1)$, as this final equation may have up to the $D+1$ terms $1, x_1, \ldots, x_1^D$ (or up to $x_1^{q-1}$ if $q \leq D$) instead of $T = I - 1$). Surprisingly (cf. Sec. 6.2) this may offer little practical improvement over $T - I = 1$.

## 3 The Family of XL Variants

When proposing XL ([20]) the authors noted that we need $m - n \geq 2$ for good performance. Which brings us the "FXL" method as the first of several XL variants.

### 3.1 FXL: Guessing as Aid to Equation-Solving

The "F" in FXL stands for "fix" ([20]). The attacker assigns random values to $f$ variables, in effect guessing at them, hoping to decrease (cf. also XL', Sec. 3.3) the degree $D$ needed for XL. After guessing, we run XL and test at the end if any solution found is valid. The complexity for $f$ variables fixed at degree $D$ is

$$C_{\text{FXL}} = q^f \left[ C_0 + E\left( \binom{n-f+D}{D},\ m\binom{n-f+D-2}{D-2} \right) \right], \tag{2}$$

where $C_0$ is a presumed small cost of collation. We will establish the worthiness of FXL by demonstrating its gains, and give some guidelines for its profitable application in Sec. 6.3. We note the fixing concept applies almost verbatim to the $\mathbf{F_4}$ and $\mathbf{F_5}$. I.e., we may also guess at a few variables before applying a Gröbner Bases method. We shall show that this can be a good idea in general.

### 3.2 XL2: Gaining Extra Equations via the $T'$ Method

This was first proposed ([22]) as an addendum to XL over GF(2), to add useful equations. Let $T'$ count the monomials that when multiplied by a given variable will still be in $\mathcal{T} = \mathcal{T}^{(D)}$. I.e. $T' = |\mathcal{T}_i'|$, where $\mathcal{T}_i' = \{\mathbf{x^b} : x_i\mathbf{x^b} \in \mathcal{T}\}$ for each $i$. Suppose $I$ is not as large as $T - D$, but $C \equiv T' + I - T > 0$ (i.e. we have enough equations to eliminate all monomials not in $\mathcal{T}_i'$), then:

1. Eliminate from the system $\mathcal{R} = \mathcal{R}^{(D)}$ the monomials not in $\mathcal{T}_1'$ first. We are then left with relations $\mathcal{R}_1$ that gives each monomial in $\mathcal{T} \setminus \mathcal{T}_1'$ as a linear combination of those monomials in $\mathcal{T}_1'$, plus $C$ equations $\mathcal{R}_1'$ with only monomials in $\mathcal{T}_1'$.
2. Repeat for $\mathcal{T}_2'$ to get the equations $\mathcal{R}_2$ and $\mathcal{R}_2'$ (we should also have $|\mathcal{R}_2'| = C$).
3. For each $\ell \in \mathcal{R}_1'$, monomial in the equation $x_1\ell = 0$ are either in $\mathcal{T}_2'$ or can be reduced (using $\mathcal{R}_2$) into $\mathcal{T}_2'$. Ditto each $x_2\ell$ ($\ell \in \mathcal{R}_2'$) and we get $2C$ new equations.

XL2 is described as a sequence of Buchberger relations by [54]. It is important it is similar to the final stage ($T'$-method) of the related XSL (extended sparse linearization, [21]) method that purports to break block ciphers with *sparse quadratic structure*, including AES. We do not analyze XSL itself here. [22] claims that most of the $2C$ equations *"are likely"* to be linearly independent, and that XL2 can be repeated for an eventual solution. We seek to clarify the heuristics below.

### 3.3 XL': Searching as the Final Step

XL' ([22]) is XL except that we come down to a system in $r$ variables and at least $r$ equations, then end by brute-force search. The total time complexity for large $q$ is

$$C_{\text{XL'}} \approx E\left(\binom{n+D}{D}, m\binom{n+D-2}{D-2}\right) + \frac{q^r D}{1-\frac{1}{q}}\binom{r+D}{D}. \quad (3)$$

The new terminations conditions are: instead of requiring $T - I \leq D$, we only require $T - I \leq \binom{r+D}{D} - r$. Note: It is usually 1-in-$q$ for any polynomial to vanish on random inputs, and we must test degree-$D$ polynomials with $r$ variables and up to $\binom{r+D}{D}$ terms. We need a suitably small $q^r$ and make some changes. This $D$ is smaller than the $D_0$ for regular XL. We will check how much smaller in Sec. 7.

### 3.4 XLF: Using the Field Relations

[17] proposes to use the field relations $x^q = x$ to advantage when $q = 2^k$:

- Consider $(x_i^2)$, $(x_i^4)$, ..., $(x_i^{2^{k-1}})$ independent variables in $K$ in addition to $x_i$.
- Equations are generated as in every other XL method, then each generated equation is raised to the second, fourth,... powers easily (since this is a linear operation) as equations in $(x_i^2), \ldots, (x_i^{2^{k-1}})$, for $k$ times as many variables *and* equations.
- That all equivalent monomials are *ipso facto* equal become new equations, which may let the algorithm execute with a lower $D$ (see Sec. 7).

### 3.5 XFL: Guessing with a Twist

Another variant proposed with the name "improved FXL" and later XFL ([17, 59]):

1. Choose $f$ ("to fix") variables. Multiply the equations by all monomials up to degree $D - 2$ *in the other* $n - f$ *variables only.*
2. Order the monomials so that all monomials of exactly degree $D$ with no "to-fix" factor comes first. Eliminate all such monomials from the top-degree block.
3. Substitute actual values for "to-fix" variables, then collate the terms and try to continue XL, re-ordering the monomials if needed, until we find at least one solution.

There are $\binom{n-f+D-1}{D}$ monomials of degree $D$ with no "to-fix" variable, so $T' = \binom{n-f+D}{D} - \binom{n-f+D-1}{D} = \binom{n-f+D-1}{D-1}$ variables remain and the complexity is:

$$C_{\text{XFL}} = C_0'' + q^f\left[C_0' + E\left(\binom{n-f+D-1}{D-1}, m\binom{n-f+D-2}{D-2} - \binom{n-f+D-1}{D}\right)\right]. \quad (4)$$

$C_0''$ the cost of the initial elimination. What happens is that the max-degree block of the elimination need not repeat with the guessing. We shall see how this does later.

## 4 Gröbner Bases Algorithms $\mathbf{F_4}$-$\mathbf{F_5}$

Gröbner Bases have come a long way since the early days of Buchberger. The reader is referred to [6, 10, 11, 40] for general theory on the topic, although the speed estimates there can be considered superseded. The most advanced implementations are detailed in [29, 30, 31]. Summaries can also be found in [2, 54], here we only give a synopsis:

0. Initialize: The original are reduced according to some (usu. Degree Reverse Lexicographic) monomial order to a system in row-echelon form.
1. Multiply/Extend: Increase the maximal degree by 1. The resulting equations are multiplied by all monomials such that the product does not exceed the maximal degree. In $\mathbf{F_5}$ the Frobenius selection criteria avoids redundant equations.
2. Linearize/Reduce: Run a Gaussian-like elimination to row-echelon form, such that every row/equation is only reduced against preceeding rows.
3. Repeat: If we do not yet have a Gröbner Basis, go to Step 1. We will find a Gröbner Bases as in $x_1 = f_1(x_2, x_3, \ldots, x_n)$, $x_2 = f_2(x_3, \ldots x_n)$, $\ldots$, maybe ending with $f_{k+1}(x_{k+1}, \ldots, x_n) = 0$ when the system variety has positive Krull dimension.

Please refer to the abovementioned articles for technical details. Lazard (cf. [40]) notes long ago that a Gröbner Basis for a set of equations $\ell_i$ may be found by a reduction on the extended version of the *Macaulay matrix* at some degree $D$. This matrix contains exactly the coefficients of the equations $\mathcal{R}^{(D)}$, and the reduction of this matrix is exactly XL. Hence [2] and [54] explains XL as a special case of Gröbner Algorithms.

## 5 Termination Conditions of XL and Gröbner Bases

How many independent equations do we get in the basic XL? Not all equations are independent: If we write $\ell_i(\mathbf{x}) = \sum_{j \le k} a_{ijk} x_j x_k + \sum_j b_{ij} x_j + c_i$, then

$$\begin{aligned}[\ell_i \ell_{i'}] &= \sum_{j \le k} a_{ijk} [x_j x_k \ell_{i'}] + \sum_j b_{ij} [x_j \ell_{i'}] + c_i \, [\ell_{i'}] \\ &= \sum_{j \le k} a_{i'jk} [x_j x_k \ell_i] + \sum_j b_{i'j} [x_j \ell_i] + c_{i'} [\ell_i],\end{aligned}$$

where $[x_j \ell_i]$ denotes the equation $x_j \ell_i(\mathbf{x}) = 0$ in the XL system, etc., i.e., two ideals spanned by each pair of $(\ell_i, \ell_j)$ intersect, hence there will be a corresponding dependency at every degree $D > 4$. We may compute the number of free equations assuming no other source of dependencies than the above:

**Proposition 1 ([24, 58]).** *If all dependencies result from $\ell_i[\ell_{i'}] = \ell_{i'}[\ell_i]$ then*

$$T - I = [t^D] \left\{ (1-t)^{m-n-1} \, (1+t)^m \right\} = \sum_{j=0}^{\infty} (-1)^j \binom{m-n-1}{j} \binom{m}{D-j}, \quad (5)$$

*for all* $D < \min(q, D_{reg})$. *Here* $D_{reg}$ *is the degree of regularity given by*

$$D_{reg} := \min\{D : [t^D]\ \left((1-t)^{m-n-1}\ (1+t)^m\right) \leq 0\}, \tag{6}$$

*and* $[t^k]\ p$ *means "*the coefficient of $t^k$ in the expansion of $p$*". E.g.* $[x^2](1+x)^4 = 6$. *This implies that the minimum* $D$ *required for the reliable termination of XL is given by*

$$D_0 := \min\{D : [t^D]\ \left((1-t)^{m-n-1}\ (1+t)^m\right) \leq D\}. \tag{7}$$

**Historical Remark:** The [58] proof was faulty and did not prove $D_0$ to be a lower bound. T. Moh ([44]) states without proof a result similar to this one. C. Diem has the first and only derivation ([24]) showing $D_0$ to be a lower bound if the *Maximum Rank Conjecture* (originally due to Fröberg, [32]) is generally valid.

**Corollary 2.** *When there are no extraneous dependencies (i.e., Eq. 5 holds), then* $D_0$ *is:* $2^n$ *if* $m = n$, $m$ *if* $m - n = 1$, *and* $\lceil (m+1)/2 \rceil$ *if* $m - n = 2$.

*Proof.* If $m - n = 0$, then $T - I = [t^D]\ ((1+t)^m/(1-t)) = \sum_{j=0}^{D} \binom{m}{D}$, which increases rapidly. It stays constant after reaching $2^m$ at $D = m$, so $D_0 \geq \max(2^m, q)$.

If $m - n = 1$, then $T - I = [t^D]\,(1+t)^m = \binom{m}{D} > D$ up to and including $D = m-1$ (whence $T - I = m > D$). Finally at $D = m$ we have $T - I = 1 < m$. The $D_0 = m - 1 = n$ of [17, 20] is due to a slightly different XL in [20].

If $m - n = 2$, then $T - I = [t^D]\ ((1-t)\,(1+t)^m) = \binom{m}{D} - \binom{m}{D-1} = \binom{m}{D-1}\frac{m-2D+1}{D}$. Obviously $T - I \leq 0$ iff $D \geq (m+1)/2$, and $T - I > D$ early on. So $D_0 \leq \lceil (m+1)/2 \rceil$. When $D$ is incremented by 1, $T - I$ increases by $\left(\binom{m}{D} - \binom{m}{D-1}\right) - \left(\binom{m}{D-1} - \binom{m}{D-2}\right) = \binom{m}{D+1} - 2\binom{m}{D} + \binom{m}{D-1}$, which starts out positive and when $D > \frac{1}{2}(m - \sqrt{m+2})$ turns negative (see below), so we only need to check the case of $D = \lceil (m+1)/2 \rceil - 1$ first, which is the last $D$ before $T - I$ decreases down to or past 0. Combinatorics texts (e.g. [52]) tell us that $\binom{2D}{D} - \binom{2D}{D-1} = \frac{(2D)!}{D!(D+1)!}$ is the Catalan number $c_D$, which satisfy $c_0 = 1,\ c_n = \sum_{j=0}^{n-1} c_j c_{n-1-j}$. Since $c_2 = 2$, $c_D$ for $D > 2$ will be the sum of $D$ positive integers, not all of them 1, so $c_D > D$. Similarly for even $m$ we have $\binom{2D+1}{D} - \binom{2D+1}{D-1} = \frac{(2D+2)!}{(D+1)!(D+2)!} = c_{D+1} > D$.

So the bounds are tight. Eqs. 5 and 7, says that as described in [20], as $m - n$ increases $D_0$ decreases, although formulas are more complicated for larger $m - n$ (Tab. 1).

**Corollary 3.** *Good approximations to* $D_0$ *for fixed small* $f = m - n$ *is given by Tab. 1.*

$D_0$ is not easily expressed as a function of $m$ (or $n$) for larger $f = m - n$. We may approximately assume that $D_0 \approx D_{reg}$. $D_0$ is then the smallest $D$ such that

$$T - I = [t^D]\ \left((1-t)^{f-1}\,(1+t)^m\right) = \sum_{j=0}^{f-1} (-1)^j \binom{f-1}{j}\binom{m}{D-j} < 0.$$

or, after dividing out $(m!)/[D!(m-D+f-1)!]$, we get this inequality in $D$:

$$\sum_{j=0}^{f-1}(-1)^j \binom{f-1}{j} \frac{D!}{(D-j)!} \frac{(m-D+f-1)!}{(m-D+j)!} \leq 0 \tag{8}$$

From Eq. 8 we can find $D_{reg}$ explicitly (and $D_0$ approximately) for $f \leq 10$ using lots of roots. We tabulate (cf. Table 1) the results for smaller $f = m - n$. This shows that the earlier estimate of $D_0 \approx \sqrt{n}$ ([20]) for small $f$ is not very good. Indeed, [24] points out for any fixed $f$, $D_0/n \to 1/2$.

**Table 1.** Relationship between $f = m - n$ and minimal degree $D_0 = D_0(m)$

| $f$ | $D_0$ (as approximate function of $m$) | 10 | 15 | 20 | 25 | 30 | 35 |
|---|---|---|---|---|---|---|---|
| 0 | $2^m$ (but only up to $q$, cf. Sec. 5) | $2^{10}$ | $2^{15}$ | $2^{20}$ | $2^{25}$ | $2^{30}$ | $2^{35}$ |
| 1 | $m$ | 10 | 15 | 20 | 25 | 30 | 35 |
| 2 | $\lceil \frac{1}{2}(m+1) \rceil$ | 6 | 8 | 11 | 13 | 16 | 18 |
| 3 | $\lceil \frac{1}{2}(m+2-\sqrt{m+2}) \rceil$ | 5 | 7 | 9 | 11 | 14 | 16 |
| 4 | $\lceil \frac{1}{2}(m+3-\sqrt{3m+7}) \rceil$ | 4 | 6 | 8 | 10 | 12 | 14 |
| 5 | $\left\lceil \frac{1}{2}(m+4-\sqrt{3m+8+\sqrt{6m^2+30m+40}}) \right\rceil$ | 3 | 5 | 7 | 9 | 11 | 13 |
| 6 | $\left\lceil \frac{1}{2}(m+5-\sqrt{5m+15+\sqrt{10m^2+50m+76}}) \right\rceil$ | 3 | 4 | 6 | 8 | 10 | 12 |

The predictions of Eq. 5 is confirmed for random quadratics $\ell_i$ by simulations due to N. Courtois ([18]) up to very high dimensions and degrees, including all the parameters listed in [17] and earlier works. The public polynomials of several PKC's including SFLASH$^{v2}$ also behaves like random polynomials at low degrees. This verifies our own simulations, which are not so extensive.

### 5.1 XL2, Gröbner Bases, and Their Relationship

Corresponding to Eq. 7 for Gröbner Bases algorithms such as $\mathbf{F_5}$ we have ([4], later [2]) is this result for semi-regular sequences of equations (i.e., no extra dependencies):

$$D_g := \min\{D : [t^D](1-t)^{m-n}(1+t)^m < 0\}. \tag{9}$$

Its resemblence to $D_0$ for XL means that some results for XL can extend directly to Gröbner Bases: We can think of this as corresponding to exactly one fewer variable, or we can think of the extra factor of $(1-t)$ to mean that the elimination is run on the highest-ranked monomials only. One variant method of XL does exactly that — the XL2 adjunct (Sec. 3.2), otherwise known as the $T'$-method. In [58, 59], it was pointed out that one can run XL2 on all variables to achieve effectively going one degree higher. [59] comments that XL2 may not repeat even if it works once. It may be possible using the original, overly optimistic estimate $(T - I < T')$ as opposed to one that focuses on the top degree monomials; indeed, we *prove* below that it is not the case for large $q$.

**Proposition 4.** *XL2 (for large q) on all variables will run when $D \geq D_g$, and will then repeat until we find a solution or prove the system self-contradictory.*

*Proof.* We need to eliminate all top-degree monomials ([59]), and can think of regular XL being run on homogenized equations with one variable assigned to represent the constant 1, and we may apply Eq. 6, we get the first half of the statement.

We know that multiplying by $(1-t)^{-1}$ represent taking the sum of coefficients of a generating function up to some degree, i.e., $g_{m,f}(D) = \sum_{d=0}^{D} g_{m,f+1}(d)$ (here $f = m - n$). We know that the zeroes of $g_{m,f}(D) := [t^D]\left((1-t)^{f-1}(1+t)^m\right)$ and $g_{m,f+1}(D)$ alternate (see below) because their dominant terms are associated Hermite functions (which form an orthogonal sequence). So once $D \geq D_g$, $g_{m,f+1}(D)$ will not become positive until $g_{m,f}(D)$ becomes non-positive also.

We have shown that XL+XL2 and $\mathbf{F_5}$ operates at the same apparent degree $D_g$. Further ([30]) the *signature*, i.e. the underlying degree, of the polynomials in the matrix/system built by $\mathbf{F_5}$ is the same as the classical Buchberger algorithm, which is the same as classical XL (this remark was also made in [2]). This is scarcely surprising, given that Imai *et al* have shown that XL2 is equal to a sequence of Buchberger-like operations. Therefore, we can think of XL+XL2 as a less polished version of $\mathbf{F_5}$.

### 5.2 FXL and Asymptotic Estimates for $D_0$

With $m = n$ equations and variables in practice the attacker would most often run the variant FXL, i.e., guesses at $f$ variables, then attempts to run XL on the remaining system. It is hence of particular interest to obtain asymptotic behavior when $m - n = f$ remains small compared to $m$ or $n$. Eq. 7 is valid only when $D < q$, but for GF($2^8$) we can cover all $m$ up to about 500, which is large enough to bring in asymptotics. This requires first asymptotically estimating a coefficient then approximating a sign-change position in the following manner via Cauchy's integral formula ([35]),

$$g_{m,f}(D) := [t^D]\left((1-t)^{f-1}(1+t)^m\right) = \frac{1}{2\pi i}\oint z^{-D-1}(1-z)^{f-1}(1+z)^m dz.$$

Standard asymptotic analysis recipes (cf. [12, 35, 57]) can be applied to find ([3]) that

$$D_{reg} = \frac{m}{2} - (h_{f-1,1})\sqrt{\frac{m}{2}} + O(1) \sim \frac{m}{2} - \sqrt{fm}. \tag{10}$$

Here $f = o(m^{1/3})$ and $h_{k,1}$ is the largest zero of the Hermite polynomial $H_k(x)$, given by Szegö ([55]) as $\sqrt{2k+1} + O(k^{-1/6})$. And when we have $f \sim cm$

$$D_{reg} \sim \left(\frac{1}{2} - \sqrt{c} + \frac{c}{2}\right) m + O(m^{\frac{1}{3}}). \tag{11}$$

via the Coalescent Saddle Point method ([3, 12]). We note that Eqs. 10 and 11 are compatible which is necessary if the asymptotics are uniform.

One consequence of the above is that that an optimal $f$ for FXL (cf. Sec. 6.3) exist, which also applies to $\mathbf{F_4}$-$\mathbf{F_5}$. Let us start with a medium-large $m = n$ (asymptotics come into play as low as in the teens), and start with the assumption that $D = m/2$, then we may compute $\lg T = 1.377m + O(\log m)$ via Stirling's formula:

$$T = \binom{n+D}{n} = \binom{3m/2}{m} = \frac{(3m/2)!}{m!(m/2)!} \sim \sqrt{\frac{3}{2\pi m}} \left( \frac{(3/2)^{3/2}}{(1/2)^{1/2}} \right)^m$$

When guessing at $f$ variables, Eq. 10 means that $n$ and $D$ goes down respectively by $f$ and roughly $\sqrt{fm}$. We find that $T$ goes down by a factor of $\approx \left( (3/2)^f \cdot 3^{\sqrt{fm}} \right)$, hence $\lg T \sim 1.377\, m - 1.585\sqrt{fm} + 0.585f$, and if we assume small $f$, $q = 2^8$, $E$ to be degree $\omega = 2 + \varepsilon$ (a Lanczos solver, see below) and $R \sim T$ then FXL has

$$\lg C_{\text{FXL}} \sim 2 \lg T + f \lg q \sim (\lg q + 1.170) f - 3.170\sqrt{fm} + 2.755m. \tag{12}$$

If Eq. 12 holds for all $f$, then $\lg C_{\text{FXL}}$ will take a minimum of $\lg C_{\text{FXL}} \approx 2.63m$ at $f \sim 0.014m$, a significant gain. However, Eq. 12 is actually valid up for small $f$, actually to only $f = o(\sqrt[3]{m})$. We may still conclude that for FXL, there is some small $\epsilon > 0$ and $\delta$ such that we should take at least $f = \delta \cdot m^{1/3-\epsilon}$ guessed variables, and we can say more since we have compatible asymptotics, for which we refer you to [59]. The result is that we should guess even more variables: For $q = 2^8$ and $\omega = 2$ the minimum occurs at around $c := f/m \sim 0.049$, when $\lg C_{\text{FXL}} \sim 2.4m$. Similarly when $\omega = \lg 7$ (Strassen blocking), the minimum $\lg C_{\text{FXL}} \sim 3.0m$ when $f \sim 0.096m$.

Even supposing that our numbers are slightly off, this shows that FXL is a better way to apply XL on non-small fields. As Gröber Bases methods theoretically and asymptotically resemble XL, the phenomena should be nearly the same. I.e., starting from $m = n$, one should guess at a very small percentage of the variables before starting the Gröbner Bases computations. Indeed, for $m = n$ and $\omega = 2.8$ (Strassen Blocking), we see that $\lg T = \lg \binom{2n}{n} \sim 2n$, hence $\lg C_{\mathbf{F_4}} \sim \lg C_{\mathbf{F_5}} \sim 5.6n$, as opposed to $4.2n$ for guessing at *one* variable, and $3.0n$ with the optimal guessing. For $\omega = 2$, the coefficients are 4, 3.0 and 2.4 respectively. This seems very natural, but has not been seen in print previously. Results for smaller $q$ can be found in [4, 58, 59]. As C. Diem points out, a critical proof in [58] is inaccurate, its results are not always valid lower bounds. However [59] shows FXL (and likewise F$\mathbf{F_4}$/F$\mathbf{F_5}$ worthwhile for all values of $(q, \omega)$.

## 6 Pragmatic Issues in XL-Related Methods

We first mention some theoretical and practical aspects of XL-related method, particularly the parameters we shall use when estimating the complexity (security level).

### 6.1 On a Pragmatic Cost of Elimination

Naive cubic-time elimination ([8]) is inadequate for large matrices, and a cost estimate $T^{\lg 7}$ (where $T$ counts the monomials) or even lower is cited in all XL articles ([15, 17, 20, 21, 22]). However, Strassen's ([53]) original $2^{\lg 7}$ algorithm does not reliably invert a known nonsingular square matrix. The XL situation is even more complex: The matrix (with $R > T$) is not square, and we want our elimination algorithm to (a) run despite the redundant rows (equations); (b) compute a useful basis for the kernel (e.g. reduced row-echelon form) if the matrix is not full-rank (i.e. $T - I = 1$). To pivot inerrantly around singular submatrices in $O(n^{\lg 7})$ is quite nontrivial ([7]). Similar caveats apply to adapting other sub-cubic matrix multiplications for equation-solving.

The best all-around result for dense matrices known to us is D. J. Bernstein's GGE (Generalized Gaussian Elimination, [9]) which computes the *quasi-inverse*, which can (method "S") solve a system of equations, and even (method "N") find a basis of the kernel of a matrix (a row-reduced echelon form)! *Assume $M$ equations, $N$ variables, and the time cost $\sim \alpha N^\omega$ to multiply two $N \times N$ matrices, then GGE uses time*

$$E_S(N, M) = \frac{2\alpha(1+\gamma)}{(2^\omega - 2)} M^{\omega-1} N + \frac{\alpha M^\omega}{(2^\omega - 1)}; \tag{13}$$

$$E_N(N, M) = \frac{2\alpha(2+\gamma)}{(2^\omega - 2)} M^{\omega-2} N^2 + \frac{4\alpha\gamma}{7} M^{\omega-1} N + \frac{\alpha M^\omega}{(2^\omega - 1)}. \tag{14}$$

Here the coefficient $\gamma = (7\alpha)/(2^\omega - 4)$. We shall look at how to do better in Sec. 6.2.

### 6.2 A Need for Sparse-Matrix Algorithms

The systems generated by XL are obviously sparse. A respected textbook on sparse matrices ([27]) remarks that in not using a matrix algorithms more tailored for the situation *"you would just be pushing milliards of zeros around"*. Moving around gigabytes full of zeroes not only slows down the computation direcly, but increases the amount of memory required. With $n = 15$, $m = 20$, $q = 2^8$ and $D = 7$ (these are practical dimensions), we have $T = 170544$ monomials and $R = 310080$ equations. A full elimination will take about 50 GB ($\approx 2^{36}$ bytes). A normal procedure ([27]) is to find block structures with graph-coloring analysis. The elimination cost is then dominated by $E(N_0, M_0)$, the elimination cost for the largest block. *The XL equations are structured such that the largest block comprise the equations with the highest degrees (and this naturally happen in $\mathbf{F_5}$).* Still, if we know that there is at most one solution, then it must be better to use Lanczos, Conjugate Gradient (CG), or Wiedemann methods, each solving an $N \times N$ system $\mathsf{M}\mathbf{x} = \mathbf{b}$ using $N$ multiplications of $\mathsf{M}$ to an $N \times 1$ vector. An $M \times N$ system ($M > N$) in Lanczos (or CG) is converted to $N \times N$ by solving $(\mathsf{M}^T\mathsf{D}\mathsf{M})\mathbf{x} = \mathsf{M}^T\mathsf{D}\mathbf{b}$ instead. Here $\mathsf{D}$ is invertible, diagonal, and suitably random. *For sparse systems with $t$ terms per equation, the expected time cost drops to order $2 + \varepsilon$:*

$$E_L(M, N) \approx (c_0 + c_1 \lg N)\, tMN. \tag{15}$$

The log-factor is because accessing memory no longer take negligible time, and tags are $\propto \lg N$ long. Lanczos, CG and Wiedmann methods all have comparable speeds. Consensus seems to peg the Wiedemann algorithm as intrinsically slower but more reliable, and to get better results Lanczos methods must be randomized which adds to the cost (cf. [8, 28, 38, 56]). Warning: *Lanczos (or CG) is known to terminate sometimes incorrectly over a finite field. Wiedemann is not known to terminate always correctly for nonsquare matrices. Proper operating conditions are not fully understood.* However, we were informed ([13]) that such methods are usable and widely used in practice.

### 6.3 Practical Parameters for Assessing FXL (for Large $q$)

Since we ultimately want reasonable security estimates, we need concrete values for $c_0$ and $c_1$ in the Lanczos estimate $E_L$. What are reasonable estimates? *We will use $c_0 = 4$ and $c_1 = \frac{1}{4}$ in Eq. 15 to arrive at complexity estimates (for Lanczos-like sparse solvers) in field multiplications. Calibrating against our own test data, we should divide the number of multiplications by roughly $2^6$ to get numbers in 3DES encryption blocks (comparable to but a little longer than AES blocks).*

Furthermore, if the dimensions become very large, then asymptotically we will eventually see $R/T$ in the hundreds. However, we may generate fewer equations ([13, 18]), e.g., via a randomly picked set of equations (taking say 20% more equations than variables) and solve ten such random systems to ensure not missing a solution. Hence it makes much more sense to assume the equations to have roughly as many equations as there are variables, and we may assume $R/T$ to be a constant on the order of "a few". Of course, for smaller dimensions, it may make more sense to run a more robust elimination. For Gröbner Bases methods, obviously $T = R$ and this is a smaller $T$ because it is only the top degree portion, but this contribution dominates the number of monomials for large $q$ and a typical case of XL/FXL anyway. With $\alpha = 7$, $\omega = \lg 7$ ([41]), we get

$$E_{\text{sparse}}(N, M) = (\frac{1}{4} \lg N + 4) \cdot MN \cdot (\text{ avg. \# of terms per equation}); \quad (16)$$

$$E_{\text{dense}}(N, M) = 51.33M^{0.8}N^2 + 65.33M^{1.8}N + 1.167M^{2.8}.$$

These numbers are for processors with enough cache only. We hear that some IBM servers do have 100+GB of RAM and a mind-boggling 512MB cache per CPU, so we assume that processing power, memory size and bandwidth all pose no problems.

## 7 Practical Security Assessment of XL Variants

Infeasibility of the cryptanalysis against SFLASH$^{v2}$ and HFE Challenge 2 as mentioned in [59] is given, using some results that we prove rigorously for semi-regular sequences.

### 7.1 Inefficiency of XL' and XLF for Small $m - n$ and Large $q$

**Proposition 5.** *The number of extra equations provided by XLF (Sec. 3.4) is given by*

$$\Delta T = k \binom{n + \lfloor D/2 \rfloor}{\lfloor D/2 \rfloor} - 1. \tag{17}$$

*Proof.* We need not track the redundant monomials explicitly as in [17]. These monomials are the degree $\leq D$ monomials in the $(x_i)$'s that can also be written as monomials of the $(x_i^2)$'s at a lower degree, copied $k$ times. So we just count monomials in the $(x_i^2)$'s of total degree $\leq D/2$, which number $\binom{n+\lfloor D/2 \rfloor}{\lfloor D/2 \rfloor}$. The final $-1$ comes from the fact that the monomial 1 is counted as duplicated $k$ times, once too many.

**Corollary 6.** *When $D < q$, XLF can be expected to work (most of the time) when*

$$[t^D] \left( (1-t)^{m-n-1} (1+t)^m \right) - \binom{n+\lfloor D/2 \rfloor}{\lfloor D/2 \rfloor} < \lceil D/2 \rceil. \tag{18}$$

Note: This is likely only asymptotically correct (extra dependencies are possible). If the elimination ends with all odd powers of $(x_1^{2^j})$ left we can still solve for $x_1$.

**Proposition 7.** *The following holds about XL' and XLF for $q$ large:*

**XL':** *When $m - n$ is 1 or 2, XL' operates if and only if $D > m - r$; if $m = n$, XL' will not run at $D = m + 1 - r$, but will at $D = m + 2 - r$ for $r$ large enough (around $r > m/2$). When $r$ is small, we need a much larger $D$, around $2^{m/r} (r!)^{1/r}$.*

**XLF:** *When $m - n \leq 2$, XLF need at least $D > n/2$ to operate.*

*Proof.* We can use the approximations $\binom{2k}{k} \approx 2^{2k}/\sqrt{\pi k}$ and $k! \approx \sqrt{2\pi k}(\frac{k}{e})^k$.

**XL':** $m = n$**:** From the description of XL' above and Eq. 5, we see that $\binom{r+D}{D} - r \geq \sum_{i=0}^{D} \binom{m}{i} > \binom{m+1}{D} + \binom{m+1}{D-2}$. Since $\binom{m}{k}$ is increasing in $m$, we need $r + D > m + 1$ which suffices for $r > m/2$. For small $r$, we need $\binom{r+D}{D} > 2^m$.

$m = n + 1$**:** Need $\binom{r+D}{D} - \binom{m}{D} \geq r$; when $r + D = m + 1$, the left hand side becomes $\binom{m}{D-1} \geq m > r$, so it does the job, and no smaller $r$ can do that.

$m = n + 2$**:** Need $\binom{r+D}{D} = \binom{r+D+1}{D} - \binom{r+D}{D-1} > \binom{m}{D} - \binom{m}{D-1}$; again $r + D = m + 1$ will do, and barely, because nothing smaller works.

**XLF:** Let $m - n = 1$. As we can presume $D$ small, we have gives $\binom{n+1}{D} < \binom{n+\lfloor D/2 \rfloor}{\lfloor D/2 \rfloor}$. At $D = n/2$ we have the left side $\approx \frac{2^{n+1}}{\sqrt{\pi n/2}}(1 - \frac{1}{n+2})$, and the right

$$\approx \frac{\sqrt{2\pi \cdot 5n/4} \cdot (\frac{5n}{4e})^{5n/4}}{\left(\sqrt{2\pi \cdot n/4} \cdot (\frac{n}{4e})^{n/4}\right) \left(\sqrt{2\pi n} \cdot (\frac{n}{e})^n\right)} = \sqrt{\frac{5}{2\pi n}} \left[ \frac{(5/4)^{5/4}}{(1/4)^{1/4}} \right]^n.$$

We see $\left\lceil \frac{(5/4)^{5/4}}{(1/4)^{1/4}} \right\rceil \approx 1.87 < 2$ and $\frac{2}{\sqrt{\pi n/2}}(1 - \frac{1}{n+2}) > \sqrt{\frac{5}{2\pi n}}$. For $m = n$ we need a higher $D$ (we can check that $D \approx 3m/4$ is needed).
Now consider $m - n = 2$. We want $\binom{n+2}{D} - \binom{n+2}{D-1} = \binom{n+1}{D} - \binom{n+1}{D-2} < \binom{n+\lfloor D/2 \rfloor}{\lfloor D/2 \rfloor}$, which we can verify to happen only when $D \leq n/2$. The LHS is about $1/D$ of what it was at $m - n = 1$, which is covered by the exponential factor $(1.87/2)^n$.

XL' (designed for GF(2)) work suboptimally on a larger field. XLF is hindered by the fact that the dependencies are multiplied along with the independent ones ([18]).

### 7.2 XFL Is Really a Space-Time Tradeoff

XFL of Sec. 3.5 appears to be an improvement, but there are important drawbacks. Essentially, for the initial elimination stage, the memory requirement is increased $(m - f)$-fold. More importantly, once the initial substitution is made, the resultant second-highest-degree block is no longer sparse and requires the equivalence of GGE (Sec. 6.1).

There is no particular reason that XFL should fail. Indeed, it is better than XL' or XLF. However we believe that FXL works better due to the availability of Lanczos.

### 7.3 Reassessing XL'/XLF/XFL Versus SFLASH$^{v2}$ and HFE Challenge 2

**Proposition 8 ([58]).** *If $2q > D \geq q$, and the system is semi-regular up to degree $D$, then $T - I = [t^D]\left((1-t)^{m-n-1}(1-nt^q)(1-t^2)^m\right)$. [Also similarly for $\mathbf{F_4}$/$\mathbf{F_5}$.]*

This is a yardstick we need for the complexity of some XL variants, and we look at how three XL variants apply to extant schemes SFLASH$^{v2}$ and HFE Challenge 2.

**Did XL Variants really break SFLASH$^{v2}$ in $2^{80}$?** To recap, SFLASH$^{v2}$ ([51]) is a NESSIE finalist. It is an instance of $C^{*-}$ ([49], descendant of $C^*$, [43])) with $K = \mathrm{GF}(2^7)$, $m = 26$ and $n = 37$, and reputed to be very fast, suitable for smart card implementations ([1]). Although the NESSIE writeup contained some extraneous private data that can be recovered ([34]), SFLASH$^{v2}$ was previously considered safe ([47]). It is claimed ([17]) that after $n$ is reduced to 26 by guessing at eleven variables, any of the variants XL', XLF, and XFL can provide a cryptanalysis within $2^{80}$ 3DES operations. None of the cryptanalysis attempts can function as given:

**XL':** [17] gives $D = 7$, $r = 5$. From Sec. 7.1, we can see that at $r = 5$, XL' should not work until $D = 92$. We actually ([18]) need $D = 93$. By trial and error, we get best result is around $r = 16$, which gives a complexity of $\sim 2^{118}$.

**XFL:** [17] gives $f = 4$, $D = 6$. Actually we see from Table 1 that $D_0 \geq 10$. With Strassen and Bernstein, we get $2^{104}$ multiplications ($2^{98}$ 3DES blocks).
**XLF:** [17] gives $D = 10$. Using Sec. 7.1 and Prop. 8 we verify that XLF only works at $D = 18$ (complexity $\sim 2^{92}$ even with Lanczos).

Reports of the demise of SFLASH$^{v2}$ is exaggerated and justifies the design decisions of Patarin *et al.* This is significant as SFLASH$^{v3}$ ([19]) is much slower with bulkier keys, and has security concerns due to unlucky choice in dimensions ([18, 26], cf. also `http://www.minrank.org/sflash/`). The best cryptanalysis is F$\mathbf{F_5}$ if it works with Lanczos (complexity $\gtrsim 2^{81}$). Else the best try is likely FXL (complexity $\sim 2^{85}$). If an attack works, it probably will be an algebraic attack resembling [26].

**Did XL Variants really break HFE challenge 2 in $2^{80}$?** HFE Challenge 2 is an HFE instance with $q = 2^4$ and $m = n = 32$. We believe that the parameters as given in [17] does not lead to cryptanalysis under $2^{80}$, after double-checking against Sec. 7.1:

**XL'** [17] gives $m = n = 32$, $D = 10$, for which $T - I = 107594213$. We can do (cf. Prop. 7) XL' using $(D, r) = (15, 19)$ or $(14, 20)$, which is very sufficient.
**XFL:** $D = 7$ and $f = 2$ ([17]) won't function (since $T - I = 2459664$). We recommend $(f, D) = (12, 6)$ with complexity $\sim 2^{97}$ 3DES blocks.
**XLF:** [17] gives $D = 10$, which we can verify not to work (as above). We need (cf. Prop. 8) all the way to $D = 23$, with a complexity $2^{112}$ even for Lanczos.

## 8 Discussions and Conclusion

With all the results we have gathered, we may tabulate the complexity in various schemes. Two points need explaining. F1$\mathbf{F_5}$ and F2XL means $\mathbf{F_5}$ guessing at *one* variable and FXL always guessing at two variables respectively. The asterisk means that we are assuming Lanczos-class speed solvers to work with $\mathbf{F_5}$, which is not a given.

**Table 2.** Time Estimates (3DES blocks): Blocking ($\omega = 2.8$) v. Lanczos ($*$: may not work)

| Attack Method | | FXL | F2XL | $\mathbf{F_5}$ | F1$\mathbf{F_5}$ | F$\mathbf{F_5}$ |
|---|---|---|---|---|---|---|
| $n = 26$ $q = 2^7$ (SFLASH$^{v2}$) | B | $2^{101}$ | $2^{104}$ | $2^{117}$ | $2^{102}$ | $2^{99}$ |
| | L | $2^{85}$ | $2^{87}$ | $2^{95}_*$ | $2^{85}_*$ | $2^{82}_*$ |
| $n = 32$, $q = 2^4$ (HFE Chal. #2) | B | $2^{97}$ | $2^{106}$ | $2^{111}$ | $2^{105}$ | $2^{93}$ |
| | L | $2^{87}$ | $2^{92}$ | $2^{98}_*$ | $2^{89}_*$ | $2^{82}_*$ |
| $n = 20$, $q = 2^8$ | B | $2^{91}$ | $2^{91}$ | $2^{109}$ | $2^{88}$ | $2^{86}$ |
| | L | $2^{78}$ | $2^{78}$ | $2^{84}_*$ | $2^{77}_*$ | $2^{74}_*$ |
| Asymptotic for big $m = n$, $q$ | B | $2^{3.0n}$ | $2^{3.86n}$ | $2^{5.6n}$ | $2^{3.86n}$ | $2^{3.0n}$ |
| | L | $2^{2.4n}$ | $2^{2.75n}$ | $2^{4.0n}_*$ | $2^{2.75n}_*$ | $2^{2.4n}_*$ |

### 8.1 Comments on the Relationship of Gröbner Bases to XL

Imai *et al* ([54]) shows XL to be variation of the $\mathbf{F}_4$-$\mathbf{F}_5$ algorithms. However, practical differences remain even if the theory of XL might be considered subsumed by Gröbner Bases. Gröbner Bases is a general and elegant mathematical theory that applies well to everything under the sun including symbolic computation. When $\mathbf{F_5}$ terminates , we should always obtain all solutions, including those in extension fields. Shamir *et al* proposed XL as a cryptanalytical tool, with one purpose: to find a known-or-conjectured-to-exist solution to a numerical set of equations. In FXL/Lanczos variant, this property is shown clearly: it is possible to find all solutions in $K$, but not in extensions of $K$.

Wiedemann and Lanczos algorithms are not suitable for computing reduced row-echelon forms; as a conclusion to XL, either works best with $T - I = 1$. In a Gaussian, we need not know $T - I$ beforehand and may come down to any number of monomials (between 1 and $\min(q-1, D)$) with no speed penalty and still terminate correctly; under Bernstein's GGE, we are penalized by the slower algorithm "N" (instead of "S", cf. Sec. 6.1–6.2); using Lanczos requires us to know $T - I$ in advance, and to run exactly that many different iterative sequences. [30, 31] seem to agree with the above assessment, and the critical step of $\mathbf{F}_4/\mathbf{F}_5$ appears to be an elimination on the top block.

The aversion to Gaussion or Generalized Gaussian elimination is also why we do not suggest XL2. We do not see how to link it reliably into a Lanczos-like sparse solver.

[4] claims that Gaussian-like elimination in $\mathbf{F_5}$ can achieve time close to Lanczos algorithms. At least, $\omega = 2$ is "plausible". We hasten to agree! It is quite plausible that one can adapt these advanced Gröbner Bases methods for Lanczos, or achieve $\omega = 2 + \varepsilon$ regardless. However, it is also plausible that one cannot, because according to [30], the elimination is severely restricted in the order of operations. We may also use guessing in $\mathbf{F_5}$, and the two methods behaves very similarly (as expected). But in this event, the two methods could be described as having largely converged. The entries that require running a Lanczos-like sparse solver with $\mathbf{F_5}$ or Fix-then- $\mathbf{F_5}$ (denoted "F$\mathbf{F_5}$") is marked with an asterisk in Tab. 2. If effectively for $\mathbf{F_5}$ we will always have $\omega$ measurably greater than 2, then these estimates are invalid. In this case FXL will eventually dominate methods that always compute a Gröbner Basis.

There is one further situation where FXL might work better, which is when we cannot hold the entire matrix of the $\mathbf{F_5}$ in memory. In turn, we can run FXL without generating the whole Macaulay matrix. All the possibilities takes further study.

### 8.2 Some Remarks on the Termination of Basic XL

Moh in [44] pointed out that Basic XL should not work if the system of equations has a positive-dimensional solution at "at infinity". It is our aim to help to clarify this often-cited remark by Moh. We thank C. Diem for pointing out Prop. 9 to us.

As above, let $\ell_1, \ldots, \ell_m \in K[x_1, \ldots, x_n]$. Let $\ell_1^h, \ldots, \ell_m^h \in K[x_0, \ldots, x_n]$ denote the homogenizations of the $\ell_i$. Let $D \in \mathbb{N}$, and let us assume (without loss of generality) that $\ell_1, \cdots, \ell_a$ have degree $\leq D$ and $\ell_{a+1}, \cdots, \ell_m^h \in K[x_0, \ldots, x_n]$ have degree $> D$. For example, for $i = 1, \ldots, a$, the $\ell_i$ might be quadratic and for $i = a+1, \ldots, m$, the $\ell_i$ might be the field equations which might have a much higher degree.

Let $V_D$ be the projective algebraic set defined by the equations

$$\ell_1^h = 0, \ldots, \ell_a^h = 0.$$

We note that if the system $\ell_1 = \cdots = \ell_a = 0$ defines a 0-dimensional algebraic set and the "set at infinity" is non-empty, the dimension of $V_D$ equals the dimension of the "set at infinity". Let $T$ and $I$ be as above. Our want to relate $T - I$ to the dimension of $V_D$.

**Proposition 9.** *If* $\dim(V_D) = r$, *then* $T - I \geq \binom{r+D}{r}$.

*Proof.* Let $\mathcal{J}$ be the homogeneous ideal defined by $\ell_1^h, \cdots, \ell_a^h$, and let $(K[x_0, \ldots, x_n]/\mathcal{J})_D$ be the $D$-th homogeneous part of the quotient ring $K[x_0, \ldots, x_n]/\mathcal{J}$.

Note that $\dim(V_D) = \dim(K[x_0, \ldots, x_n]/\mathcal{J}) - 1$ (since $V_D$ is projective), and

$$T - I = \dim((K[x_0, \ldots, x_n]/\mathcal{J})_D)$$

(see [24–Section 4] for a derivation of this formula). We can go from $K$ to any field extension $L$ without changing the numbers $T, I$ and $\dim(V_D)$. By going to a sufficiently large extension $L/K$, we can apply the Noetherian Normalization Theorem in the form of [39–Theorem 2.2]. We obtain that the ring $L[x_0, \ldots, x_n]/(\mathcal{J})$ contains a polynomial ring $L[y_0, \ldots, y_r]$ such that the images of the $y_i$ are linear combinations of the $x_i$, hence

$$\dim_K((K[x_0, \ldots, x_n]/\mathcal{J})_D) = \dim_L((L[x_0, \ldots, x_n]/(\mathcal{J}))_D) \geq$$
$$\dim_L((L[y_0, \ldots, y_r])_D) = \binom{r+D}{r}$$

Note that the proposition implies in particular that if $r \geq 1$, then $T - I \geq D + 1$, and if $r \geq 2$, then $T - I \geq \frac{(D+2)\cdot(D+1)}{2}$.

If $T - I \leq D$, then XL will find a univariate polynomial, whereas if $T - I$ is greater than this number it will *usually* not find such a polynomial. The interpretation of the proposition above is thus that *usually* XL does not terminate if $\dim(V_D) > 0$. We would like to stress that $V_D$ is the variety defined by $\ell_1^h = 0, \ldots, \ell_a^h = 0$, the equations $\ell_{a+1}^h = 0, \ldots, \ell_m^h = 0$ of higher degree are disregarded.

Let us now set $V$ as the projective algebraic set defined by *all* equations $\ell_1^h = 0, \ldots, \ell_m^h = 0$. Then we have the following good news ([44–Lemma 2] is a corollary).

**Corollary 10.** *If* $V = \emptyset$ *or* $\dim(V) = 0$, *then XL terminates for some* $D$.

*Proof.* Let $\mathcal{J}$ be the homogeneous ideal defined by $\ell_1^h, \ldots, \ell_m^h$. We now have

$$T - I = \dim((K[x_0, \ldots, x_n]/\mathcal{J})_D)$$

for all $D$ (where $T$ and $I$ are defined with respect to $D$).

Now, under our assumptions on $V$, there exists a $\tilde{D} > 0$ such that for $D \geq \tilde{D}$, $\dim(K[x_0, \ldots, x_n]/\mathcal{J}_D)$ is equal to the number of non-trivial solutions (counted with multiplicities) of the system $\ell_1^h = 0, \ldots, \ell_m^h = 0$. This is one of the statements of Hilbert theory. It is essentially proven in [36–I, §7]. It follows that *for some $D > 0$, one has $T - I \leq D$, and the algorithm finds a univariate polynomial.*

Cor. 10 does not apply to XSL (cf. 3.2) or any other method in which an entire ideal $\mathcal{I} = K[\ell_1, \ldots, \ell_m, p_1, p_2, \ldots, p_\kappa]$ is not used (the $p_i$'s are extra polynomials added by the attacker). It also does not say what $D$ is. Even if it always works, it may be slow.

When XSL ([21]) proposes to break AES (and Serpent). The more optimistic claims of cryptanlysis in $2^{87}$ or $2^{100}$ is based on XSL applied to the Murphy-Robshaw structural equations ([45, 46]) of AES. It is claimed ([17, 21]) that XSL can sidestep the objections of [44] because M-R equations are formed with techniques similar to Sec. 3.4, and the final ("$T'$-method") stage is similar to XL2. But these variant methods may work correctly or not, independently of Cor. 10.

### 8.3 A Conclusion

With all the analysis given here we hope to have done a reasonable job in covering various aspects of XL. In passing we may have rehabilitated the reputation of SFLASH$^{v2}$ to some extent. In conclusion: XL is a simplified version of current Gröbner Bases algorithms. Some prior claims about XL variants were clearly too ambitious, and sometimes unrealistic claims were put forward. Yet, the invention of XL (and particularly FXL) is clearly an advance, justifying the insights of Courtois, Klimov, Patarin and Shamir. We hope we have evaluated the capabilities of XL algorithms in a more rational and pragmatical manner. Still, much remains to be done in the practical arena. One important item is to settle the question of the validity of XSL.

The results of this work along with [59] should go some ways to show that FXL is the best XL variant, and the principle extends to $\mathbf{F_4}/\mathbf{F_5}$. We hope that this study will lead to better equation-solving methods based on F$\mathbf{F_5}$ (or F$\mathbf{F_4}$). On the theoretical side, there are also a couple of things that can use a little further study. One is the identification of situations where $\mathbf{F_5}$ (and XL/FXL) will work substantially better or much worse than the [4] bound. Another is a correct way to implement sparse matrix arithmetic so that $\mathbf{F_5}$ can reliably run with a solver with Lanczos-like speeds. While the MAGMA project ([42]) has an implementation of $\mathbf{F_4}$ that is very well optimized, even faster than Faugère's own $\mathbf{F_5}$, it is not yet pushing the limits of what such a solver can do. This is an area that can still be exciting and practically useful.

## Acknowledgements

The authors would like to thank Dr. Nicolas Courtois and Dr. Claus Diem for helpful discussions, and the first author would like to dedicate this to the 65th birthday and imminent retirement of his father, Prof. Wei-Zhe Yang of National Taiwan University.

## References

1. M. Akkar, N. Courtois, R. Duteuil, and L. Goubin, *A Fast and Secure Implementation of SFLASH*, PKC 2003, LNCS 2567, pp. 267–278.
2. G. Ars and J.-C. Faugère, H. Imai, M. Kawazoe, and M. Sugita, *Comparison of XL and Gröbner Bases Algorithms over Finite Fields*, ASIACRYPT'04, LNCS 3329, pp. 338–353.
3. M. Bardet, *Étude des systèmes algébriques surdéterminés. Applications aux codes correcteurs et à la cryptographie.*, Ph.D. thesis, Université Paris 6, 2004.
4. M. Bardet, J.-C. Faugère, and B. Salvy, *Complexity of Gröbner Basis Computations for Regular Overdetermined Systems*, INRIA Rapport de Recherche No. 5049; a slightly modified preprint is accepted by the International Conference on Polynomial System Solving.
5. M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang, *Asymptotic Complexity of Gröbner Basis Algorithms for Semi-regular Overdetermined Systems over Large Fields*, manuscript.
6. B. Barkee *et al*, *Why You Cannot Even Hope to Use Gröbner Bases in Public-Key Cryptography*, J. Symbolic Computations, 18 (1994), pp. 497–501.
7. J. R. Bunch and J. E. Hopcroft, *Triangular Factorizations and Inversion by Fast Matrix Multiplication*, Math. Computations, 24 (1974), pp. 231–236.
8. R. Burden and J. D. Faires, *Numerical Analysis, 7th ed.*, PWS-Kent Publ. Co., 2000.
9. D. Bernstein, *Matrix Inversion Made Difficult*, preprint, stated to be superseded by a yet unpublished version, available at `http://cr.yp.to`.
10. L. Caniglia, A. Galligo, and J. Heintz, *Some New Effectivity Bounds in Computational Geometry*, AAECC-6, 1988, LNCS 357, pp. 131–151.
11. L. Caniglia, A. Galligo, and J. Heintz, *Equations for the Projective Closure and Effective Nullstellensatz*, Discrete Applied Mathematics, 33 (1991), pp. 11–23.
12. C. Chester, B. Friedman, and F. Ursell, *An Extension of the Method of Steepest Descents,* Proc. Camb. Philo. Soc. 53 (1957) pp. 599–611.
13. D. Coppersmith, private communication.
14. N. Courtois, *The Security of Hidden Field Equations (HFE)*, CT-RSA 2001, LNCS 2020, pp. 266–281.
15. N. Courtois, *Higher-Order Correlation Attacks, XL Algorithm and Cryptanalysis of Toyocrypt*, ICISC '02, LNCS 2587, pp. 182–199.
16. N. Courtois, *Fast Algebraic Attacks on Stream Ciphers with Linear Feedback*, CRYPTO'03, LNCS 2729, pp. 177-194.
17. N. Courtois, *Algebraic Attacks over* $\mathrm{GF}(2^k)$*, Cryptanalysis of HFE Challenge 2 and SFLASH*$^{v2}$, PKC '04, LNCS 2947, pp. 201-217.
18. N. Courtois, private communication.

19. N. Courtois, L. Goubin, and J. Patarin, *SFLASH$^{v3}$, a Fast Asymmetric Signature Scheme*, preprint available at `http://eprint.iacr.org/2003/211`.
20. N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, EUROCRYPT 2000, LNCS 1807, pp. 392–407.
21. N. Courtois and J. Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, ASIACRYPT 2002, LNCS 2501, pp. 267–287.
22. N. Courtois and J. Patarin, *About the XL Algorithm over GF(2)*, CT-RSA 2003, LNCS 2612, pp. 141–157.
23. J. Daemen and V. Rijmen, *The Design of Rijndael, AES - The Advanced Encryption Standard.* Springer-Verlag, 2002.
24. C. Diem, *The XL-algorithm and a Conjecture from Commutative Algebra*, ASIACRYPT'04, LNCS 3329, pp. 323–337 and private communication.
25. W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Trans. Info. Theory, vol. IT-22, 6 (1972), pp. 644-654.
26. J. Ding and D. Schmidt, *Cryptanalysis of SFlash$^{v3}$*, `eprint.iacr.org/2004/103`.
27. I. S. Duff, A. M. Erismann, and J. K. Reid, *Direct Methods for Sparse Matrices*, published by Oxford Science Publications, 1986.
28. W. Eberly and E. Kaltofen, *On Randomized Lanczos Algorithms*, Proc. ISSAC '97, pp. 176–183, ACM Press 1997.
29. J.-C. Faugére, *A New Efficient Algorithm for Computing Gröbner Bases (F4)*, Journal of Pure and Applied Algebra, 139 (1999), pp. 61–88.
30. J.-C. Faugère, *A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)*, Proceedings of ISSAC 2002, pp. 75-83, ACM Press 2002.
31. J.-C. Faugère and A. Joux, *Algebraic Cryptanalysis of Hidden Field Equations (HFE) Cryptosystems Using Gröbner Bases*, CRYPTO 2003, LNCS 2729, pp. 44-60.
32. R. Fröberg, An Inequality for Hilbert Series of Graded Algebras, Math. Scand. 56(1985) pp. 117-144.
33. M. Garey and D. Johnson, *Computers and Intractability, A Guide to the Theory of NP-completeness*, W. H. Freeman New York 1979.
34. W. Geiselmann, R. Steinwandt, and T. Beth, *Revealing 441 Key Bits of* SFLASH$^{v2}$, 3rd NESSIE Workshop, 2002.
35. H.-K. Hwang, *Asymptotic estimates of elementary probability distributions*, Studies in Applied Mathematics, 99:4 (1997), pp. 393-417.
36. R. Hartshorne, *Algebraic Geometry*, Springer-Verlag, 1977.
37. A. Kipnis and A. Shamir, *Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization*, CRYPTO'99, LNCS 1666, pp. 19–30.
38. B. LaMacchia and A. Odlyzko, *Solving Large Sparse Linear Systems over Finite Fields*, CRYPTO'90, LNCS 537, pp. 109–133.
39. S. Lang, *Algebra* (3rd edition), Addison-Wesley, 1993.
40. D. Lazard, *Gröbner Bases, Gaussian Elimination and Resolution of Systems of Algebraic Equations*, EUROCAL '83, LNCS 162, pp. 146–156.
41. C. McGeoch, *"Veni, Divisi, Vici"*, Appearing in the "Computer Science Sampler" column of the Amer. Math. Monthly, May 1995.
42. The MAGMA project, University of Sydney, see `http://magma.maths.usyd.edu.au/users/allan/gb`
43. T. Matsumoto and H. Imai, *Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption*, EUROCRYPT'88, LNCS 330, pp. 419–453.
44. T. Moh, *On The Method of XL and Its Inefficiency Against TTM*, available at `http://eprint.iacr.org/2001/047`

45. S. Murphy and M. Robshaw, *Essential Algebraic Structures Within the AES*, Crypto 2002, lncs 2442, pp. 1–16.
46. S. Murphy and M. Robshaw, *Comments on the Security of the AES and the XSL Technique*, preprint available from the authors `http://www.isg.rhul.ac.uk/~sean/`
47. *NESSIE Security Report, V2.0*, available at `http://www.cryptonessie.org`
48. J. Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms*, Eurocrypt'96, lncs 1070, pp. 33–48.
49. J. Patarin, L. Goubin, and N. Courtois, $C^*_{-+}$ *and HM: Variations Around Two Schemes of T. Matsumoto and H. Imai*, Asiacrypt'98, lncs 1514, pp. 35–49.
50. J. Patarin, N. Courtois, and L. Goubin, *QUARTZ, 128-Bit Long Digital Signatures*, CT-RSA 2001, lncs 2020, pp. 282–297. Update at `http://www.cryptonessie.org`
51. J. Patarin, N. Courtois, and L. Goubin, *FLASH, a Fast Multivariate Signature Algorithm*, CT-RSA 2001, lncs 2020, pp. 298–307. Update with SFLASH$^{v2}$ available at `http://www.cryptonessie.org`
52. R. Stanley, *Enumerative Combinatorics*, vol. 1, second printing 1996; vol. 2 in 1999. Both published by Cambridge University Press, Cambridge.
53. V. Strassen, *Gaussian Elimination is not Optimal*, Numer. Math. 13 (1969) pp. 354-356.
54. M. Sugita, M. Kawazoe, and H. Imai, *Relation between XL algorithm and Groebner Bases Algorithms*, preprint, `http://eprint.iacr.org/2004/112`. Part of this merged with [2].
55. G. Szegö, *Orthogonal Polynomials, 4th ed.*, publ.: American Math. Society, Providence.
56. D. Wiedemann, *Solving Sparse Linear Equations over Finite Fields*, IEEE Transaction on Information Theory, v. IT-32 (1976), no. 1, pp. 54–62.
57. R. Wong, *Asymptotic Approximations of Integrals*, Academic Press, San Diego, 1989.
58. B.-Y. Yang and J.-M. Chen, *Theoretical Analysis of XL over Small Fields*, ACISP 2004, lncs 3108, pp. 277-288.
59. B.-Y. Yang, J.-M. Chen, and N. Courtois, *On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis*, ICICS '04, lncs 3269, pp. 401-413. Formerly titled *Exact and Asymptotic Behavior of XL-Related Methods*.