Algebraic Attacks over $GF(2^k)$, Application to HFE Challenge 2 and Sflash-v2

Nicolas T. Courtois

Axalto Cryptography Research & Advanced Security,
36-38 rue de la Princesse, BP 45, 78430 Louveciennes Cedex, France
http://www.nicolascourtois.net
courtois@minrank.org

Abstract. The problem MQ of solving a system of multivariate quadratic equations over a finite field is relevant to the security of AES and for several public key cryptosystems. For example Sflash, the fastest known signature scheme (cf. [1]), is based on MQ equations over $GF(2^7)$, and Patarin's 500 \$ HFE Challenge 2 is over $GF(2^4)$. Similarly, the fastest alleged algebraic attack on AES due to Courtois, Pieprzyk, Murphy and Robshaw uses a MQ system over $GF(2^8)$.
At present very little is known about practical solvability of such systems of equations over $GF(2^k)$. The XL algorithm for Eurocrypt 2000 was initially studied over $GF(p)$, and only recently in two papers presented at CT-RSA'02 and ICISC'02 the behaviour of XL is studied for systems of equations over $GF(2)$. In this paper we show (as expected) that XL over $GF(2^k)$, $k > 1$ (never studied so far) does not always work very well. The reason is the existence of additional roots to the system in the extension field, which is closely related to the remark made by Moh, claiming that the XSL attack on AES cannot work. However, we explain that, the specific set of equations proposed by Murphy and Robshaw already contains a structure that removes the problem. From this, we deduce a method to modify XL so that it works much better over $GF(2^k)$. In addition we show how to break the signature scheme Sflash-v2 recently selected by the European consortium Nessie, by three different methods derived from XL. Our fastest attack is in 2^{58}. All the three attacks apply also to HFE Challenge 2, and our best attack is in 2^{63}.

Key Words: Multivariate quadratic equations, MQ problem, overdefined systems of multivariate equations, XL algorithm, Gröbner bases, algebraic attacks on AES, XSL, Murphy-Robshaw equations on AES.

1 Introduction

In the perpetual search for hard problems on which to base cryptographic security, there is a growing interest in so called "multivariate problems". These problems are usually NP-hard. In terms of scalability of the systems, the best problems are those for which all known attacks are exponential: it is then sufficient to increase slightly the parameter sizes, to keep up with progress in the attacks, or with an increase in the speed of computers. One of such problems is the problem MQ, of solving a system of multivariate quadratic equations over a small finite field. Several public key cryptosystems based on MQ have been proposed, for example the HFE family [30, 9]. In this paper we study generic attacks that solve the underlying MQ problem independently of the existence of the trapdoor. They apply also to random quadratic equations.

At Crypto'99, Shamir and Kipnis present a surprising method called relinearization for solving overdefined systems of multivariate quadratic equations. They point out that, if such a system of equations is overdefined (much more equations than needed), then it can be solved much faster than expected. Subsequently, at Eurocrypt 2000 [32], Courtois, Klimov, Patarin and Shamir, present a new algorithm called XL, (and also FXL) that can be seen as an improved version of relinearization.

From [32] and still at present, very little is known about the exact complexity and behaviour of XL. Initially in [32] it was studied mainly over $GF(p)$. Recently a lot of interest emerged in solving MQ systems over $GF(2)$ and $GF(2^k)$, due to the Courtois-Pieprzyk method to attack AES by such means [15, 26]. At CT-RSA 2002 Courtois and Patarin study the XL algorithm over $GF(2)$ and show it works much better than expected from [32] or from the naive criticism of it published on the internet [24]. At ICISC 2002, Courtois studies the extension of XL to equations of degree higher than 2, and again demonstrates that it works very well, allowing to cryptanalyse the stream cipher Toyocrypt, see [7]. The object of this paper is to study rather MQ over fields of the form $GF(2^k), k > 1$. Such equations appear for example in the signature schemes Flash, Sflash and Sflash-v2 published at CT-RSA 2002, out of which Sflash-v2 has been selected by Nessie (in company of ECDSA and RSA-PSS). Also, in the fastest known alleged attack on AES due to Courtois-Pieprzyk and Murphy-Robshaw [15, 26], the equations are quadratic over $GF(2^8)$.

2 Notation and Conventions Used in this Paper

The MQ Problem

In this paper we consider the problem of solving a system of m multivariate quadratic equations with n variables over a finite field $GF(q)$. We use very similar notations than in [32] and [14]. The input variables are denoted by x_i and belong to $GF(q)$ with $q = 2^k$. The equations are denoted by l_i and are quadratic (which means they can also include linear and constant terms). Our system to solve will be:

$$\mathcal{A} : \begin{cases} l_1(x_1, \ldots x_n) & = 0 \\ & \vdots \\ l_m(x_1, \ldots x_n) & = 0 \end{cases}$$

Given m, n, q we call MQ the problem of finding one (not necessarily all) solutions to such a system chosen at random. Typically in cryptographic applications, k can be between 4 and 8 and m, n can between 26 and 1600 (for AES, see [15, 26]). The MQ problem is NP-hard, see [20].

Remark: In XL description in [32, 14] the powers of variables are taken in $GF(q)$, i.e. reduced modulo q to the range $1, \ldots, q-1$, because of the equation $x_i^q = x_i$ of the finite field $GF(q)$. Thus if $q = 2$ there would be no powers of x_i bigger than 1. For us it makes no difference, as in all cases studied in this paper, we have $q \geq 16$ and we will never generate or manipulate equations of degree equal or bigger than $q-1$.

Instances of MQ that Will Be Used in This Paper

If $m > n$ the system is said to be overdefined. Similarly as in [32, 14], we will see that for a fixed n, the bigger is m, the more easy becomes the MQ problem. If $m < n$ the system is said to be underdefined, and efficient algorithms for the underdefined MQ has been studied in [5]. In general, following [32, 14], we expect that the hardest case of MQ is when $m \approx n$.

In practice, if we have a system with $n > m$, as in the Sflash public key [12], we will start by fixing some variables to arbitrary values, get a system with $m \geq n$, and the try to solve it. (When over $GF(2^k)$, it is unclear if one can take advantage from the initial $n > m$, cf. [5].)

For all our MQ systems we will always insure/assume that **the system has one and unique solution**, we refer to Section 4.1 or to the end of Section 5.1 to see why it is very important. To have one unique solution happens frequently in cryptographic applications of MQ, and it is also the average number of solutions of a random MQ with $m = n$. Moreover, in practice, for systems that have several solutions, we can always reduce to a system having one solution, by guessing a few variables.

Manipulating the Equations

Because the right hand of all our equations is always 0, it is very useful to identify a multivariate polynomial and an equation that says it is equal to 0. Thus the equation $l_i(x_1, \ldots x_n) = 0$ can be simply called the equation l_i, and the equation $x_1 \cdot l_2(x_1, \ldots x_n) = 0$ can be called simply $x_1 l_2$.

We say that the equations of the form $\prod_{j=1}^{k} x_{i_j} \cdot l_i = 0$, with all the i_j being pairwise different, are of type $x^k l$, and we call $x^k l$ the set of all these equations. For example the initial equations $\mathcal{A}$ are of type l. We observe that each solution x that satisfies all the equations l_i, also does satisfy all the equations of type $x^k l$, for any $k \geq 0$. Similarly we denote by x^k the set of all terms of degree exactly K, $\prod_{j=1}^{K} x_{i_j}$. By extension we define $x^0 = \{1\}$, the constant monomial.

Let $D \in \mathbb{N}$. We consider all the polynomials $\prod_j x_{i_j} \cdot l_i$ of total degree $\leq D$. Let $\mathcal{I}_D$ be the set of equations they span. $\mathcal{I}_D$ is the linear space generated by all the $x^k l,\ 0 \leq k \leq D-2$. We have $\mathcal{I}_D \subset \mathcal{I}$, $\mathcal{I}$ being the ideal spanned by the l_i We call $\mathcal{T}$ the set of monomials, including the constant monomial, that appear in all the equations of $\mathcal{I}_D$, $\mathcal{T} = \bigcup_{i=0}^{D} x^i$.

3 The Basic Principle of XL

Let D be the parameter of XL algorithm. Following [32, 14]:

Definition 3.0.1 (The XL algorithm). Execute the following steps:

1. **Multiply:** Generate all the products $\prod_{j=1}^{k} x_{i_j} \cdot l_i \in \mathcal{I}_D$ with $k \leq D-2$.
2. **Linearize:** Consider each monomial in the x_i of degree $\leq D$ as a new variable and perform Gaussian elimination on the equations obtained in 1. The ordering on the monomials must be such that all the terms containing one variable (say x_1) are eliminated last.
3. **Find $\mathbf{x_1}$:** Assume that step 2 yields at least one univariate equation in the powers of x_1. Solve this equation over the finite fields (e.g., with Berlekamp's algorithm). There may be several roots.
4. **Recover the other variables:** For each root x_1 substitute it to the expanded equations and, directly from the Gaussian reduction done in step 3, find the values of all the other monomials, in particular for all the other variables x_i.

4 The Necessary Condition for XL to Work

We will always assume $q = 2^k, k > 1$. We also always assume $D < q-1$, because we will have $q \geq 16$ and and D will remain quite small (XL is exponential in D). The XL algorithm consists of multiplying the initial m equations l_i by all possible monomials of degree up to $D-2$, so that the total degree of resulting equations is D. With the notations introduced above, this set of equations is called $\mathcal{I}_D$. Let R be the number of equations generated in $\mathcal{I}_D$ and T be the number of all monomials. When $D < q-1$ we have:

$$T = |\mathcal{T}| = \sum_{i=0}^{D} |x^i| = \sum_{\lambda=0}^{D} \binom{n+\lambda-1}{\lambda} = \binom{n+D}{D}$$

$$R = |\mathcal{I}_D| = m\left(\sum_{\lambda=0}^{D-2} \binom{n+\lambda-1}{\lambda}\right) = m\binom{n+D-2}{D-2}$$

It is likely that not all of these equations are linearly independent, and we denote by $Free$ the exact dimension of $\mathcal{I}_D$. We have $Free \le R$. We also have necessarily $Free \le T$.

The basic principle of XL is the following: one monomial in T can be generated in many different ways when different equations are multiplied by different monomials. Therefore T grows slower than R and for some D we will have $R \ge T$. Then we expect that $Free \approx T$, as obviously it cannot be bigger than T. In [32], when $Free \ge T - D$, it is possible to obtain one equation with only one variable x_1, and XL will succeed. (However in [14] two improved versions of XL are introduced: XL' and XL2, that will work when $Free < T - T'$, for some T' that may be substantially bigger then D.)

Simplified Analysis of XL from [32]

In Section 6 of [32], R is evaluated as $R = m \cdot \frac{n^{D-2}}{(D-2)!}$ and T is evaluated as $\frac{n^D}{D!}$. The authors state that "if most of the equations are linearly independent" then XL will succeed as long as $R \ge T$, which gives that: $m \ge \frac{n^2}{D(D-1)}$, and thus they obtain the (approximative) bound $D \ge \frac{n}{\sqrt{m}}$.

4.1 General Theory and Moh's Comments on XL

In [24], Moh states that "From the theory of Hilbert-Serre, we may deduce that the XL program will work for many interesting cases for D large enough". According to [23], in XL we always have $Free \le T - \alpha$. and when D is sufficiently big, we have $Free = T - \alpha$. Here α is the number of solutions to the system, including not only the solutions when $x_i \in Gf(q)$, but also when the x_i lie in an algebraic extension of the field $GF(q)$, or projective solutions (points at infinity). Thus, on the one side, under our condition that our system has one and unique solution, and if there is no projective solutions or in an extension field, XL should work and for D large enough we should have $Free = T - 1$. On the other side, this condition is necessary, and when the system has several solutions, $Free = T - 1$ is never achieved and the basic XL cannot work. Thus, in Section 4 of [24], Moh shows an interesting example on which the XL always fails. However:

- For XL over $GF(2)$, it is shown in [14] that this kind of counter-example cannot occur, because of the added equations $x_i^2 = x_i$ that make that the system has no points at infinity, and the additional solutions in the algebraic closure of $GF(2)$ are excluded.
- In this paper we work over $GF(2^k), k \ne 1$ and we will face this problem. In Section 6 we will see that XL will not work well when $m = n$,

then in Section 7 we will present a new version of XL, called XLF, that will work even in this case. (In addition, we will see that in practice, if 2^k is not too big, and only then, two other already known versions of XL can also circumvent this problem.)

5 Important Remarks About XL Algorithm over $GF(2^k)$

Let $Free$ be the dimension of $\mathcal{I}_D$, i.e. the maximum number of equations that are linearly independent. Very little is known about the value of $Free$ for $D \geq 3$. In the paper that describes XL, the authors demonstrate that XL works with a series of computer simulations over $GF(127)$ (and some more are given in the extended version of the paper [32]). In [14, 7] the authors study the XL algorithm over $GF(2)$. They do many computer simulations and are able to predict the exact value $Free$ obtained in these simulations. In this paper we will do the same for XL over $GF(2^k), k > 1$.

5.1 The Behaviour of XL - Upper Bounds

In general it is not always possible to have $Free = R$. In many cases the equations generated by XL are not all linearly independent. One reason for this is that $Free$ cannot exceed T, as the equations lie in a linear space spanned by all the T monomials. We have therefore always

$$Free \leq \min(T, R)$$

Moreover, it is possible to see that if the system is not contradictory, and has one solution, then:

$$Free \leq \min(T - 1, R)$$

This can be shown by contradiction: if $Free = T$ then by elimination of $T - 1$ non-constant monomials, some liner combination of the given equations will be 1, and if there is a solution to these equations, by substituting it, we get $0 = 1$.

5.2 The Behaviour of XL - Interesting Cases

As we will see in the present paper, the behaviour of XL over $GF(2^k)$, when k is not too small, (e.g. $k = 7$) is very similar to the general behaviour of XL over a big field $GF(p)$ studied in details (with many computer simulations) in [32]:

- XL works very well for (even slightly) overdefined systems of equations, i.e. when m exceeds n by even a small value, cf. Appendix A.
- However when $m \approx n$, and as long as the XL parameter D is smaller than the cardinal of the field, it is possible to see that XL does **not** work very well for systems of quadratic equations over $GF(2^k)$.

A different behaviour is observed for XL over a very small finite field (such as $GF(2)$ or $GF(3)$): XL works much better and there is no “problem” at all when $m \approx n$. Detailed explanation and many computer simulations for this case are given in [14] and in the appendix of [7].

6 Our Computer Simulations on XL

In all our simulations we pick a random system of linearly independent quadratic (non-homogenous) equations $y_i = f_i(x_1, \ldots, x_n)$ and pick a random input $x = (x_1, \ldots, x_n)$. Then we modify the constants in the system in order to have a system that has a solution (and gives 0 in x). The system solve is then of the form $l_i(x_0, \ldots, x_{n-1}) = 0$, for $i = 1, \ldots m$.

In Appendix A we show that for overdefined systems of equations over $GF(2^k)$, i.e. when $m > n + \varepsilon$, XL works very well. Below we study the hard case, when $m \approx n$.

6.1 Simulations on XL over $GF(2^k)$ when $m = n$

Table 1. XL over $GF(2^7)$ for $m = n$

n	2	2	2	3	3	3	3	3	4	**4**	4	4	4
m	2	2	2	3	3	3	3	3	4	**4**	4	4	4
D	2	3	4	2	4	6	7	8	4	**5**	10	15	**16**
R	2	6	12	3	30	105	168	252	60	**140**	1980	2860	12240
T	6	10	15	10	35	84	120	165	70	**126**	1001	1365	4845
$Free$	2	6	11	3	27	76	112	157	54	**110**	985	1349	4829
$\frac{Free}{T-D}$	0.50	0.86	1.00	0.38	0.87	0.97	0.99	1.00	0.82	**0.91**	0.99	0.99	1.00
$Success$			OK					OK					OK

Legend:
- n number of variables.
- m number of equations.
- D we generate equations of total degree $\leq D$ in the x_i.
- R number of equations generated (independent or not).
- T number of monomials of degree $\leq D$.
- Free number of linearly independent equations among the R equations.
- $\diamond$ Note: XL will work when $Free \geq T - D$.

It is very interesting to observe the column in bold characters: though already for $D = 5$ XL gives $R > T$ and therefore it could work, it will not work until we have $D = 16$. The difference is quite big: the complexity of the attack grows exponentially in D.

We see that for $m = n$ and over $GF(2^7)$ the XL algorithm works very poorly. In [32], for simulations over $GF(127)$, it appears that the minimum degree is $D = 2^n$. We observe the same here. The reason for this is, following [32], that for $m = n$ the system has many solutions not only in the base field, but also in the algebraic closure.

It is interesting to see that basic XL over $GF(2^7)$ becomes impractical already for $m = n = 5$: in this case, doing XL with $D = 2^5 = 32$ would give a complexity of about 2^{49}, more than exhaustive search in $2^{7 \cdot 5} = 2^{35}$. Later we will improve XL to handle such systems much faster.

6.2 Simulations on XL over $GF(2^k)$ when $m = n + \varepsilon$

We will see that, similarly as in [32], the behaviour of XL will dramatically improve when m becomes slightly bigger than n. We do not longer need $D = 2^n$ and XL works about as soon as R becomes larger than T.

Table 2. XL over $GF(2^7)$ for $m = n + \varepsilon$ (notations as for Table 1)

n	4	4	4	4	4	4	5	5	5	5
m	4	4	5	5	6	6	6	6	7	7
D	15	16	4	5	3	4	4	5	3	4
R	2860	12240	75	175	30	90	126	336	42	147
T	1365	4845	70	126	35	70	126	252	56	126
$Free$	1349	4829	65	125	30	69	111	246	42	125
$\frac{Free}{T-D}$	0.99	1.00	0.98	1.03	0.94	1.05	0.91	1.00	0.79	1.02
$Success$		OK		OK		OK		OK		OK

7 XLF - New Version of XL for $m \approx n$ and $GF(2^k)$

In Section 6.1 we saw that XL does not work very well when $m = n$ and over a large field $GF(2^k)$. From the analysis done in [32], we expect that this is due to existence of many additional solutions to our system of equations that lie in an extension field. In this section we introduce a new version of XL, called XLF, designed specifically to handle this problem over fields $GF(2^k)$. XL stands for multiply (X) and linearize (L), the new method is called XLF, which stands for multiply (X) and linearize (L) and apply Frobenius mappings (F). The basic idea of XLF is borrowed from the Murphy-Robshaw representation of AES [26]. Each variable x that appears in the system of equations will be duplicated k times, instead of x_i, we will have k variables denoted by $(x_i), (x_i^2), (x_i^4), \ldots, (x_i^{2^k-1})$. Each equation $0 = \sum_{ij} \alpha_{ij} x_i x_j$ will be also duplicated k times: we will write: $0 = \sum_{ij} \alpha_{ij}^2 (x_i^2)(x_j^2)$ etc. After doing XL expansion we get k times as many equations of degree D and k times as many variables as in the regular XL execution. Then we add some new equations that relate the new variables to each other. For example, we add $k \cdot n$ quadratic equations as follows: for each i we have $(x^2) = (x) \cdot (x)$ up to $(x) = (x^{2^{k-1}}) \cdot (x^{2^{k-1}})$. If $D \geq 4$ we have also kn equations of type $(x^4) = (x) \cdot (x) \cdot (x) \cdot (x)$ etc. Since the equations we added are only equalities between monomials, we may as well identify these monomials, which is equivalent to counting less monomials. In the extended version of this paper we give a precise list of all the monomials that are identified, and formulas to compute the resulting reduced number of monomials T.

7.1 Comparing XLF and XL

It is easy to see that by this simple trick, all the solutions with $x_i \notin GF(2^k)$ will be removed, because they cannot satisfy the added equations.

We conjecture that XLF will work as long as R becomes somewhat bigger than T in the ordinary XL, (for example twice as big). This belief is motivated by the paper [14] where it is shown that the equations of the field $GF(2)$ make XL always work as long as $R > 1.1T$.

XLF is expected to work where the original XL fails, as for $m \approx n$ XL does not work well when $R > T$, as shown in Section 6. XLF uses k times as many equations, and k times as many monomials as XL. We expect therefore that the complexity of XLF will be only about k^ω bigger than the expected complexity of XL (if the XL itself does not work). Indeed, our simulations (Table 3) show that XLF works very well when XL fails, i.e. even when $m = n$.

Table 3. XLF algorithm over $GF(2^7)$ for $m = n$ (notations as for Table 1).

n	2	2	2	3	3	3	3	3
m	2	2	2	3	3	3	3	3
D	2	3	4	2	3	4	5	6
R	14	42	84	21	84	210	420	735
T	22	50	64	43	113	176	323	449
$Free$	14	42	61	21	84	168	315	445
$\frac{Free}{T-D}$	0.70	0.89	1.02	0.51	0.76	0.98	0.99	1.00
$Success$			OK					OK

n	4	4	4	4	4	5	5	5	5	5	5
m	4	4	4	4	4	5	5	5	5	5	5
D	4	5	6	7	8	3	4	5	6	7	8
R	420	980	1960	3528	5880	210	735	1960	4410	8820	16710
T	386	778	1226	2066	2976	346	736	1618	2843	5153	8128
$Free$	350	742	1218	2058	2970	210	630	1505	2800	5110	8120
$\frac{Free}{T-D}$	0.92	0.96	0.999	0.999	1.00	0.61	0.86	0.93	0.99	0.99	1.00
$Success$					OK						OK

We see that XLF behaves much better than XL for solving systems of equations over $GF(2^k)$ with $m = n$. For example, with XL, we need $D = 2^5 = 32$ to solve a system of 5 equations with 5 unknowns, while with XLF $D = 8$ is enough. This is an important improvement, because the complexity of both XL and XLF is very much the same with an important factor that is exponential in D.

7.2 Relation with Algebraic Attacks on AES

The idea of XLF algorithm introduced in this paper is closely related to the question of feasibility of an algebraic attack on AES [15]. In the Murphy-Robshaw representation of AES, see [26] for details, the equations are written over GF(256), and have the same structure as in XLF: for each variable (x) there is a variable that is always equal to (x^2), and for each equation, the square of it also present in the system.

These equations may be combined with the Courtois-Pieprzyk XSL attack on AES, (XSL is different from XL and beyond the scope of this paper). From the values of R and T obtained in XSL it seems that, if sufficiently many equations are linearly independent, AES 128 bits would be broken in about 2^{100}, see [15, 26]. However, on a web page entitled "AES is not broken" [25], T.T.Moh unwillingly acknowledges that the XL algorithm will work, but objects for XSL and AES as follows: " new considerations of the XL method to the smallest field $GF(2)$ with the well-known trick of adding the equations $x_i^2 + x_i = 0$ to make the the component at infinity empty to satisfy the requirement of our Proposition 2" (...) "Note that this trick can not be used for the AES situation, since the corresponding equations would be $x_i^{256} + x^i = 0$, the degrees 256 would be too high for practical purpose."

This is very interesting, because as far as we know, it is the **only** somewhat mathematically founded argument that have been put forward so far, suggesting that the Courtois-Pieprzyk-Murphy-Robshaw attack in 2^{100} might not work on AES. Yet as we have seen above, this argument is void: the structure of equations makes that each of the variables must lie in $GF(256)$. This excludes additional roots in extension fields that would make the attack fail. Moreover, it is also easy to see that with such equations there will be no points at infinity: if we homogenise the equations $(x_i^2) = (x_i) * (x_i)$ with a new variable (a), we get $(a) * (x_i^2) = (x_i) * (x_i)$, and then if $a = 0$, all the x_i will be 0, which is a contradiction.

Consequences for AES: Results on XL certainly not prove that the XSL attack on AES works. Yet the Moh argument saying it shouldn't work does not apply at all to this specific Courtois-Pieprzyk-Murphy-Robshaw system of equations. More generally, this paper shows that it is in general risky, and difficult to predict whether an algebraic attack will or will not work. We have seen that for (somewhat) deeply mathematical reasons XL does not work very well for Sflash. Yet, as we will see later, a subtle and finally quite minor modification of XL, such as XLF or XL', is enough to make it work and break Sflash.

8 New Attacks on Sflash

In this section we present three new methods that allow to break Sflash in less than the Nessie security requirement of 2^{80} Triple-DES computations.

8.1 Applying XLF to Sflash

In Sflash we have $n = 37$ and $m = 26$. Equations are quadratic over $GF(2^7)$. We fix 11 arbitrary variables to 0 and still expect to have on average one solution. Then we apply XLF, the new version of XL. For $D =$

7 we have $R = 7 \cdot 4417686$ and $T \approx 7 \cdot 4272048$. Though XL does certainly fail here, we expect that XLF may work. For $D = 7$ the complexity would be about $T^\omega \approx \mathbf{2^{67}}$. Even if we were too optimistic, and XLF works only for $D = 10$, then we still have an attack in $T^\omega \approx 2^{83}$ CPU clocks which is less than 2^{80} triple-DES computations required by Nessie.

8.2 Another Attack on Sflash Using XL' Method from [14]

In this section we present yet another and even simpler method to break Sflash. Instead of XL, we apply the XL' algorithm from [14]. With classical XL, for $D = 7$ we have $R = 4417686$ and $T = 4272048$, however in practice, and $Free$ does not take the value $\geq T - D$ for a very long time. This makes XL fail so far. Still, as shown by all simulations of Section 6.1, $Free$ remains very close to $T - D$, and from this we expect that the XL' version of XL described in [14] will work. We have $n = 26$ and $m = 26$. We count all the monomials contain **only** the first 5 variables: let T' be their number, we have $T' = \binom{5+D}{D} = 792$. It seems very likely that the rank, usually close to $T - D$, will be at least $T - T' + 5$. Then we are able to eliminate **all** the monomials that contain **any** of the remaining $n - 5 = 26 - 5 = 21$ variables, and get a system of 5 equations of degree $D = 7$ with 5 variables, with $T' = 792$ monomials. Such a system can be solved by exhaustive search in about $2^{7 \cdot 5} \cdot 792 \cdot 5 \approx 2^{47}$. The total complexity of the attack will be $\left(2^{47} + T^\omega\right) \approx \mathbf{2^{58}}$ CPU clocks.

8.3 Another Attack on Sflash with Modified FXL

In this attack we will use the idea of FXL from [32]: guess values of few variables in Sflash, solve the system by XL, and then solve the system by XL. FXL leads very quickly to an overdefined system of equations and from [32] and following our experiments done in Section 6.2, we expect that after fixing a few variables XL will work.

Moreover, we will be able to do only once most of the Gaussian reduction that in FXL is done each time, which will give better results over basic FXL from [32]. We proceed as follows:

1. We start with MQ with $m = n = 26$ and over $GF(2^7)$.
2. We fix $f = 4$ variables (this is the optimal choice we have found).
3. We have 22 variables said of "type a" and $f = 4$ variables "of type b".
4. We multiply all the equations by all the products of degree up to $D = 6$ of the variables of "type a".
5. The number of equations is $R = 26\binom{22+D-2}{D-2} = 388700$.
6. In these equations we will eliminate all monomials of degree exactly $D = 6$ in the variables of "type a". Their number is exactly $T' = \binom{22+D-1}{D} = 296010$. They do not depend on the variables of "type b", and can be eliminated once for all.

7. Thus we get $R - T' = 92690$ equations that are of degree $D - 1 = 5$ in the variables of "type a".
8. If we fix a random value for the four variables of "type b", then we get a system of $R - T' = 92690$ equations with $T'' = \binom{22+5}{5} = 80730$ monomials that is sufficiently defined, as $9269 > 80730$.
9. We expect that if the guess for the four variables of "type b" is correct, then the system has a solution and the rank of this system is at most $80730 - 1$. However if the guess is wrong, we expect the system to be contradictory and the rank to be 80730.
10. We expect that on average exactly one guess will be correct.
11. The complexity to find the right values for the four variables of "type b" with Strassen's version of the Gaussian reduction is about:
$$2^{7\cdot 4} \cdot 7/64 \cdot (80730)^{\log_2(7)} \approx \mathbf{2^{71}}.$$

Remark: It is possible to see that the matrices generated in our attacks are somewhat sparse and that they can probably still be (slightly) improved by using sparse linear algebra.

9 Application of Our Attacks to HFE Challenge 2

The HFE Challenge 2 was published by Patarin in the extended version of [30], with a price of 500 \$. In the extended version of this paper we apply exactly "as they are" our 3 attacks from Section 8 Results are given in Table 4 and our best attack on HFE Challenge 2 gives 2^{63}.

10 Conclusion and Perspectives

The problem MQ of solving a set of multivariate quadratic equations over a finite field arises in cryptography (allowing to propose new cryptographic primitives), but also in cryptanalysis (for example for AES). In this paper we have studied the XL algorithm over $GF(2^k)$. We show that it works very well for overdefined equations and fails when $m \approx n$. Then we present XLF, a modified version of XL that works also in this case.

Using XLF, and also with two other versions of XL known as XL' and FXL, we present three new attacks on Sflash, a signature scheme accepted by the European Nessie consortium. All these three new attacks are faster than 2^{80}, and the fastest requires about 2^{58} CPU clocks. They also apply to Patarin's 500 \$ HFE Challenge 2, and the best gives 2^{63}.

In our results, one can notice that XLF is not the best method to break Sflash and HFE Challenge 2. This is because 2^k is still not too big. It is possible to see that, when 2^k is very big, XLF, introduced in this paper, will be **the only method known** to solve efficiently systems of quadratic equations over $GF(2^k)$ and with $m = n$. To summarize:

Table 4. Summary of the results of this paper.

	XL from [32]	XLF - new	XL' from [14]	improved FXL
Sflash-v2	2^{282}	2^{67}	$\mathbf{2^{58}}$	2^{71}
Sflash-v3 [13], $m=56$	2^{458}	2^{110}	2^{102}	2^{100}
HFE Challenge 2	2^{122}	2^{76}	2^{70}	$\mathbf{2^{63}}$
General MQ, $m \approx n, k$ big	fails	**works**	fails	fails

In Appendix A of this paper we show that, as in [14], we succeed to predict perfectly the behaviour of XL for $D < 6$, and this is sufficient to cryptanalyse current versions of Sflash and HFE Challenge 2. In general, the asymptotic behaviour of XL can be studied by the theory of Gröbner bases, see [17, 18, 16, 2]. We conjecture that complexity of solving MQ systems over a finite field with $m \approx n$ must grow exponentially with n, and even for equations over $GF(2)$, the easiest case, it can be shown that applying Buchberger algorithm to ideals generated in XL has single exponential worst case complexity, see [16] or [2].

Consequences for Sflash and HFE. We did not exhibit any structural weakness of these schemes. We simply showed that the proposed parameter sizes are insufficient for the hardness of the generic one-way problem. These schemes will resist all the attacks described in the present paper if we increase parameters m and n. Thus in Table 4 above we see that the latest updated version Sflash-v3 from [13] remains very secure.

Potential consequences for other algebraic attacks such as XSL attack on AES. We showed that for systems of low degree equations over fields $GF(2^k)$, it is not hard to avoid additional solutions in the algebraic extension or at infinity, that would make algebraic attacks fail. The Frobenius-based transformation method (with adding new variables and new equations), inspired by [26] and developed in this paper, may be of independent interest: it can potentially be applied to various systems of equations solved by various methods. For example equations can be derived from a block cipher, to be later solved by XSL-type method [15]. This simple trick (not needed in [15] nor in [26]) can transform an attack that does not work, into an attack that does work, while increasing the size of equations only k times.

Note: The extended version of this paper is available from the author.

References

1. Mehdi-Laurent Akkar, Nicolas Courtois, Louis Goubin, Romain Duteuil: *A Fast and Secure Implementation of Slash,* PKC 2003, LNCS 2567, Springer, pp. 267-278.
2. B. Barkee, D. C. Can, J. Ecks, T. Moriarty, R. F. Ree: *Why You Cannot Even Hope to use Gröbner Bases in Public Key Cryptography: An Open Letter to a Scientist*

Who Failed and a Challenge to Those Who Have Not Yet Failed, in Journal of Symbolic Computation 18, 1994, S. 497-501

3. Don Coppersmith, Shmuel Winograd: "Matrix multiplication via arithmetic progressions"; J. Symbolic Computation (1990), 9, pp. 251-280.
4. Nicolas Courtois, Magnus Daum and Patrick Felke: *On the Security of HFE, HFEv- and Quartz,* PKC 2003, LNCS 2567, Springer, pp. 337-350.
5. Nicolas Courtois, Louis Goubin, Willi Meier, Jean-Daniel Tacier: *Solving Underdefined Systems of Multivariate Quadratic Equations,* PKC 2002, LNCS 2274, Springer, pp. 211-227.
6. Nicolas Courtois: *The security of Hidden Field Equations (HFE)*; Cryptographers' Track Rsa Conference 2001, LNCS 2020, Springer, pp. 266-281.
7. Nicolas Courtois: *Higher Order Correlation Attacks, XL algorithm and Cryptanalysis of Toyocrypt,* ICISC 2002, LNCS 2587, pp. 182-199, Springer. An updated version (2002) is available at `http://eprint.iacr.org/2002/087/`.
8. Jacques Patarin, Nicolas Courtois, Louis Goubin: *C*-+ and HM - Variations around two schemes of T. Matsumoto and H. Imai*; Asiacrypt'98, Springer.
9. Jacques Patarin, Louis Goubin, Nicolas Courtois: *Quartz,* **1**28*-bit long digital signatures*; Cryptographers' Track Rsa Conference 2001, LNCS 2020, pp.282-297, Springer. See [10] for the updated Quartz specification.
10. Jacques Patarin, Louis Goubin, Nicolas Courtois: *Quartz,* **1**28*-bit long digital signatures*; An updated version of Quartz specification. available at `http://www.cryptosystem.net/quartz/`
11. Jacques Patarin, Louis Goubin, Nicolas Courtois: *Flash, a fast multivariate signature algorithm*; Cryptographers' Track Rsa Conference 2001, LNCS 2020, pp. 298-307, Springer.
12. Nicolas Courtois, Louis Goubin and Jacques Patarin: Second updated version of Sflash specification (Sflash-v2). Available at `http://www.cryptosystem.net/sflash/`
13. Nicolas Courtois, Louis Goubin and Jacques Patarin: SFLASHv3, a fast asymmetric signature scheme. New, third version of Sflash specification (Sflash-v3), proposed after this paper was written. Available on `eprint.iacr.org/2003/211/`.
14. Nicolas Courtois and Jacques Patarin, *About the XL Algorithm over* $GF(2)$, Cryptographers' Track RSA 2003, LNCS 2612, pages 141-157, Springer 2003.
15. Nicolas Courtois and Josef Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, Asiacrypt 2002, LNCS 2501, pp.267-287, Springer, a preprint with a different version of the attack is available at `http://eprint.iacr.org/2002/044/`.
16. Magnus Daum: *Das Kryptosystem HFE und quadratische Gleichungssysteme über endlichen Körpern,* Diplomarbeit, Universität Dortmund, 2001. Available from `daum@itsc.ruhr-uni-bochum.de`
17. Jean-Charles Faugère: *A new efficient algorithm for computing Gröbner bases* (F_4), Journal of Pure and Applied Algebra 139 (1999) pp. 61-88. See `www.elsevier.com/locate/jpaa`
18. Jean-Charles Faugère: *A new efficient algorithm for computing Gröbner bases without reduction to zero (F5),* Workshop on Applications of Commutative Algebra, Catania, Italy, 3-6 April 2002, ACM Press.
19. Jean-Charles Faugère: Report on a successful attack of HFE Challege 1 with Gröbner bases algorithm F5/2, announcement that appeared in `sci.crypt` newsgroup on the internet in April 19th 2002.

20. Michael Garey, David Johnson: *Computers and Intractability, a guide to the theory of NP-completeness*, Freeman, see in particular p. 251.
21. Henri Gilbert, Marine Minier: *Cryptanalysis of SFLASH.* Eurocrypt 2002, LNCS 2232, Springer, pp. 288-298, 2002.
22. Antoine Joux, Jean-Charles Faugère: *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases*, Crypto 2003, LNCS 2729, pp. 44-60, Springer.
23. Mireille Martin-Deschamps, private communication, University of Versailles.
24. T.T. Moh: *On The Method of XL and Its Inefficiency Against TTM*, invited talk at the American Mathematical Society regional meeting at the University of Notre Dame, April 8, 2000. available at `http://eprint.iacr.org/2001/047/`.
25. T.T. Moh: *On The Courtois-Pieprzyk's Attack on Rijndael*, September 18 2002, available at `http://www.usdsi.com/aes.html`.
26. S. Murphy, M. Robshaw: *Essential Algebraic Structure within the AES,* Crypto 2002, LNCS 2442, Springer.
27. NESSIE Portfolio of recommended cryptographic primitives, available at `www.cosic.esat.kuleuven.ac.be/nessie/deliverables/decision-final.pdf`
28. NESSIE Security Report, revised final version 2.0, available at `https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/D20-v2.pdf`
29. Jacques Patarin: *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*; Crypto'95, Springer, LNCS 963, pp. 248-261, 1995.
30. Jacques Patarin: *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymm. Algorithms,* Eurocrypt'96, Springer, pp. 33-48.
31. Adi Shamir, Aviad Kipnis: *Cryptanalysis of the HFE Public Key Cryptosystem*; In Advances in Cryptology, Proceedings of Crypto'99, Springer, LNCS.
32. Adi Shamir, Jacques Patarin, Nicolas Courtois, Alexander Klimov, *Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations*, Eurocrypt'2000, LNCS 1807, Springer, pp. 392-407.

A More Computer Simulations - Predicting the Behaviour of XL

In this section we will show that XL works very well for even slightly overdefined systems of equations over $GF(2^k)$, i.e. when m exceeds n by even a small value. Moreover, we will show, as in [14], how to predict the behaviour of XL, and this prediction will in many cases remain valid also when $m = n$. (the case $m \approx n$ is studied in section 6).

As before, in these simulations we pick a random system of linearly independent quadratic (non-homogenous) equations $y_i = f_i(x_1, \ldots, x_n)$ and pick a random input $x = (x_1, \ldots, x_n)$. Then we modify the constants in the system to have a system that has (at least) one solution x.

A.1 The Behaviour of XL over $GF(2^k)$ for $D = 3$

We have always $Free \le \min(T-1, R)$. We have done various computer simulations with $D = 3$ and in our simulations, for $D = 3$, we have always $Free = \min(T-1, R)$ or $Free = \min(T-1, R) - 1$.

In the following table we fix n and try XL on a random system of m linearly independent equations with growing m and with a fixed D.

Table 5. XL over $GF(2^7)$ for $D = 3$ (notations as for Table 1)

n	10	10	10	10	10	20	20	20	20	20
m	10	15	20	25	26	20	40	60	80	85
D	3	3	3	3	3	3	3	3	3	3
R	110	165	220	275	286	420	840	1260	1680	1785
T	286	286	286	286	286	1771	1771	1771	1771	1771
$Free$	110	165	220	275	285	420	840	1260	1680	1770
$Expected$	110	165	220	275	285	420	840	1260	1680	1770
$\frac{Free}{T-D}$	0.39	0.58	0.77	0.96	1.00	0.24	0.48	0.71	0.95	1.00
$Success$					OK					OK

A.2 The Behaviour of XL over $GF(2^k)$ for $D = 4$.

When $D = 4$ we do not have $Free = \min(T, R)$ anymore.

Table 6. XL over $GF(2^7)$ for $D = 4$ (notations as for Table 1)

n	5	5	5	5	10	10	10	10
m	5	6	7	8	10	15	17	18
D	4	4	4	4	4	4	4	4
R	105	126	147	168	660	990	1122	1188
T	126	126	126	126	1001	1001	1001	1001
$Free$	95	111	125	125	615	885	986	1000
$Expected$	95	111	125	125	615	885	986	1000
$\frac{Free}{T-D}$	0.76	0.91	1.02	1.02	0.62	0.89	0.99	1.00
$Success$				OK				OK

We see that for $D = 4$ most of the equations are linearly independent. We observed that we have always:

$$\text{For } D = 4, \quad Free = \min\left(T - 1, R - \binom{m}{2}\right).$$

The fact that $Free = R - \binom{m}{2}$ when $R - \binom{m}{2} \le T$, means that, in all cases, there are $\binom{m}{2}$ linear dependencies between the equations in R. As in [14], we are able to explain the origin (and the exact number) of these linear dependencies: Let l_i be the equations names (not expanded, just written as "l_i"), and let $[l_i]$ denote the expanded expression of these equations as quadratic polynomials. Then we have:

$$l_i[l_j] = [l_i]l_j$$

For each $i \neq j$, the above equation defines a linear dependency between the equations of XL. This explains the $\binom{m}{2}$ dependencies.

Example: For example if $l_1 = x_1x_3 + x_4$ and $l_5 = x_2x_1 + x_4x_7$ then the notation $l_1[l_5] = [l_1]l_5$ denotes the following linear dependency between the $l_ix_jx_k$:

$$l_1x_2x_1 + l_1x_4x_7 = l_5x_1x_3 + l_5x_4.$$

A.3 The Behaviour of XL over $GF(2^k)$ for $D = 5$.

Following the method from [14] and used in the previous chapter, we will try to predict the exact number of linearly independent equations that will be obtained for $D = 5$. First of all, we have the $\binom{m}{2}$ linear dependencies of type $l_i[l_j] = [l_i]l_j$ that are the same that existed for $D = 4$. In addition we have dependencies like:

$$l_i[l_j]x_k = [l_i]l_jx_k$$

It gives $n \cdot \binom{m}{2}$ dependencies. By inspection we check that for $D = 5$ we are unable to generate any more dependencies. From the above, we expect that:

$$\text{For } D = 5, \quad Free = \min\left(T - 1, R - (n+1)\binom{m}{2}\right).$$

Is that all the linear dependencies ? Apparently yes.

Table 7. XL over $GF(2^7)$ for $D = 5$ (notations as for Table 1)

n	5	5	5	10	10	10
m	5	6	7	10	15	16
D	5	5	5	5	5	5
R	280	336	392	2860	4290	4576
T	252	252	252	3003	3003	3003
$Free$	220	246	250	2365	3002	3002
$Expected$	220	246	250	2365	3002	3002
$\frac{Free}{T-D}$	0.88	0.98	1.00	0.79	0.99	1.00
$Success$			OK			OK

All our simulations confirm the above formula.

A.4 The Behaviour of XL over $GF(2^k)$ when $D \geq 6, \ldots$

As in [14], it is possible to continue and give formulas for $Free$ when $D = 6$ etc. These formulas are expected to predict the behaviour of XL for any D for overdefined systems with $m > n + \varepsilon$. The results given here are very similar than for fields $GF(2)$ in [14], except that in [14] the formulas work also when $m = n$, which is the hard case here.

The exact formula for all D is unknown. This formula is probably not very simple, due to entanglement of linear dependencies: so far we only subtracted linear dependencies, yet for a larger D dependencies among these dependencies will appear, etc. Apparently for XL over $GF(2)$ the exact number of linearly independent equations can be computed from the work of Jean-Charles-Faugère [17, 18], extending the so called Buchberger criteria, however we do not know if the problem is solved for XL over $GF(2^k), k > 1$.