New Variants of Algebraic Attacks Based on Structured Gaussian Elimination

Satrajit Ghosh and Abhijit Das
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, India
satrajit,abhij@cse.iitkgp.ernet.in

1 Introduction

In algebraic cryptanalysis, we express the encryption transform of a cipher as an overdefined system of multivariate polynomial equations in the bits of the plaintext, the ciphertext and the key, and then solve that system for the key bits from some known plaintext/ciphertext pairs. In general, solving such systems over finite fields is an NP-Complete problem. However, when the multivariate system is overdefined, some reasonable algorithms are known [1, 2, 3, 4, 5, 6, 7]. The XL_SGE algorithm [8] has been recently proposed to improve the complexity of the XL attack [4] by using structured Gaussian elimination (SGE) [9] during the expansion phase of XL. In this paper, we establish that XL_SGE suffers from some serious drawbacks. To avoid this problem, we propose three variants of XL_SGE, based upon partial monomial multiplication, handling of columns of weight two, and deletion of redundant equations. Our modified algorithms have been experimentally verified to be superior to XL_SGE.

We are given a sparse and consistent system $\mathbb{A}$ over $GF(2)$ of multivariate equations, some of which are quadratic and the rest of which are linear. Such systems are available from block ciphers like AES.

eXtended Linearization (XL)

In addition to the initial system $\mathbb{A}$, a degree bound D is also supplied as an input to XL [4].

Algorithm 1: Extended Linearization (XL) of multivariate systems

1. **Multiply:** Generate the new system $\mathbb{B} = \bigcup_{0 \leq k \leq D-d_{max}} X^k \mathbb{A}$, where X^k stands for the set of all monomials of degree k, and d_{max} is the maximum degree of the initial system.
2. **Linearize:** Consider each monomial in the variables x_i of degree $\leq D$ as a new variable, and perform Gaussian elimination on the system $\mathbb{B}$. The ordering of the monomials must be such that all the terms containing single variables (like x_1) are eliminated last.
3. **Solve:** Assume that Step 2 yields at least one univariate polynomial equation in some variable x_1. Find the roots of this equation in the underlying finite field.
4. **Repeat:** Simplify the equations, and repeat the process to solve for the other variables.

Structured Gaussian Elimination (SGE)

Algorithm 2 describes one iteration of structured Gaussian elimination (SGE) [9].

Algorithm 2: Structured Gaussian Elimination (SGE)

1. Delete columns of weight 0 and 1.
2. Delete rows of weight 0 and 1.
3. Delete rows of weight 1 in the light part. After Step 2 and Step 3, update column weights.
4. Delete redundant rows.

A Heuristic Improvement of XL

The problem with XL is that the size of the system increases drastically with the increase in the degree bound D. Many linearly dependent equations are generated during the expansion process (Step 1) in XL. The equations generated by XL are very sparse. Moreover, the statistics of the systems obtained in XL (for $D = 2$) reveal that the columns of the generated systems can be distinguished as heavy-weight and light-weight. These observations lead us to propose a new heuristic (XL_SGE) [8] to reduce the number of linearized equations in XL. In XL_SGE, the intermediate systems are reduced using structured Gaussian elimination (SGE). The reduced systems are multiplied with monomials to get systems of higher algebraic degrees. XL_SGE uses only the first three steps of SGE.

Algorithm 3: Extended Linearization with Structured Gaussian Elimination (XL_SGE)

1. Expand the initial system $\mathbb{A}$ up to degree $d = 2$ using XL to obtain a linearized system $\mathbb{A}'$. Make a copy of the linearized system $\mathbb{A}'$ as $\mathbb{B}$.
2. Apply structured Gaussian elimination (SGE) on $\mathbb{A}'$ with avalanche-control parameter K to obtain a reduced system of equations $\mathbb{A}''$ of degree d.
3. Multiply each equation in $\mathbb{A}''$ by each monomial of degree 1 to get a system $\mathbb{A}'''$ of degree $d + 1$. Append the equations of $\mathbb{A}'''$ to $\mathbb{B}$. $\mathbb{B}$ now has equations of degrees $\leq d + 1$. Rename $\mathbb{A}'''$ as $\mathbb{A}'$.
4. If the degree of the system of equations $\mathbb{B}$ is D, end the process. Otherwise, go to Step 2 with d incremented by 1.

XL_SGE controls excessive reduction of intermediate systems due to avalanche effects by using a heuristic parameter K during the application of SGE. More specifically, the i-th row and the j-th column are eliminated if and only if the following three conditions are satisfied: (i) the j-th column has weight 1, (ii) the (i, j)-th entry is non-zero (1, to be precise), and (iii) the weight of the i-th row is at least K.

2 Improvements of XL_SGE

XL_SGE is designed to reduce the size of the final solvable system in comparison with XL. However, there are many instances where this size reduction is not substantial. Our experiments reveal that SGE on $\mathbb{A}'$ for $d = 2$ yields sizable reduction in the system size. Subsequently, for $d \geq 3$, SGE progressively loses effectiveness in bringing down the system size. This is the expected behavior of XL_SGE.

To ensure reduction of system sizes by SGE for all degrees of $\mathbb{A}'$, two possibilities are explored. First, we investigate how variables of column weight one may reappear in the system. Second, we modify SGE to work even when all variables have column weights ≥ 2.

- **Partial monomial multiplication:** Carefully skipping certain monomial multiplications during the expansion stage has some benefits. First, fewer equations are generated, and second, SGE may again discover variables of column weight one. On the darker side, generation of fewer equations may adversely affect the rank profile of the expanded system. If too many monomial multiplications are not skipped, we hope not to encounter a big trouble with the rank profile. Therefore, two important issues are of relevance in this context: which monomial multiplications would be skipped, and how many.

- **Deletion of variables with weight more than one:** Suppose that a variable z appears in $t \geq 2$ equations in an expanded system. If we add one of these equations to the remaining $t-1$ equations, the column weight of z reduces to one, so SGE (Algorithm 2) can remove this variable in Step 1. This, however, increases the weight of these $t - 1$ equations. This increase in row weights may increase weights of certain columns. That is, an effort to forcibly eliminate z may stand in the way of the elimination of other variables. However, if $t = 2$, this processing of z followed by the removal of the only equation containing z does not increase the total weight of the system. Still, the density (average weight per row or column) of the system increases (since one equation and one variable are now removed), but the expanded systems, particularly if large, are expected to absorb this problem without sufficient degradation of the performance of XL_SGE.

XL_SGE with Random Monomial Multiplication (XL_SGE-2)

As a first attempt, we skip monomial multiplications randomly, and the amount of skipping is governed by a probability $p \in (0, 1]$. More precisely, each equation is multiplied by each monomial of degree one with probability p (and skipped with probability $1 - p$). If $p = 1$, we have the original XL_SGE algorithm. For $p < 1$, we expect more size reduction than XL_SGE.

XL_SGE-2 accepts as input the initial system of equations $\mathbb{A}$, a degree bound $D \in \mathbb{N}$, the avalanche-control parameter $K \in \mathbb{N}$, and a multiplication probability $p \in (0, 1]$.

Algorithm 4: XL_SGE with Random Monomial Multiplication (XL_SGE-2)

1. Expand the initial system $\mathbb{A}$ up to degree $d = 2$ using XL to obtain a linearized system $\mathbb{A}'$. Make a copy of the linearized system $\mathbb{A}'$ as $\mathbb{B}$.
2. Apply structured Gaussian elimination (SGE) on $\mathbb{A}'$ with avalanche-control parameter K to obtain a reduced system of equations $\mathbb{A}''$ of degree d.
3. Multiply each equation in $\mathbb{A}''$ by each monomial of degree 1 with probability p (that is, with probability $1 - p$, a multiplication is skipped) to obtain a system $\mathbb{A}'''$ of degree $d + 1$. Append the equations of $\mathbb{A}'''$ to $\mathbb{B}$. $\mathbb{B}$ now contains equations of degrees up to $d + 1$. Rename $\mathbb{A}'''$ as $\mathbb{A}'$.
4. If the degree of the system of equations $\mathbb{B}$ is D, end the process. Otherwise, go to Step 2 with d incremented by 1.

If we get a full-rank (or close-to-full-rank) system for a particular D, we solve that system. Otherwise, we increase the degree bound D, and run XL_SGE-2 again to reduce the rank deficit.

The multiplication probability p has been heuristically chosen in our experiments. We have worked with several fixed values of p in different layers (degrees d of $\mathbb{A}'$). From our experimental experiences, we recommend values of $p \geq 0.5$. A slight modification in the above algorithm for XL_SGE-2 is also studied. In this variant, monomial multiplications are randomly skipped even in Step 1 (that is, since the very beginning of the expansion process).

Another possibility is to use different probabilities in different layers of multiplication. We study two models for varying p with the degree d of $\mathbb{A}'$. In the first model, we take $p_1 = 1 - \frac{1}{d+1}$. For this choice, we initially restrict the expansion of the system. If the initial restriction leads to large rank deficits, we progressively remove the restriction on the growth of the system. In the second model, we take the gradually decreasing sequence of probabilities $p_2 = \frac{D-d}{D-d+1}$. Initially, the system size is small, so we can afford the system to grow at this stage. As d increases, $\mathbb{A}'$ becomes increasingly large, and restricting the growth of the system gradually controls the eventual growth of the system. Note also that higher-degree monomials appear in the linearized system from a larger number of sources. Hence, more restriction in the growth is required to generate more variables with column weight one as d increases.

Column-weight Two Reduction

The original SGE procedure (Algorithm 2) can be modified so as to remove columns of weights two or more. In order that the rank profile of the expanded system does not deteriorate too much, we have experimented with deletion of columns of weight two only.

Algorithm 5: Structured Gaussian Elimination with Column-weight Two Reduction (SGE′)

1. Delete columns of weight 0 and 1.
2. Delete columns of weight 2: If a column has weight 2, delete one equation corresponding to that variable. Substitute that equation in the other equation, and delete the column.
3. Delete rows of weight 0 and 1.
4. Delete rows of weight 1 in the light part. After Steps 2–4, update column weights.

Although this heuristic modification of SGE seems to be effective, in the current form it does not work very well. One must not use Algorithm 5 to reduce the initial quadratic system (available after

Step 1 of XL_SGE or XL_SGE-2), since random systems at this stage exhibit the tendency of losing all quadratic variables. Using the modified SGE for all $d \geq 3$ sometimes shows good performance. But the general observation is that the system suffers from drastic reduction in size (a form of avalanche effect) resulting in degraded rank profile and demanding a large number of iterations (that is, large values of D). It appears that the modified SGE procedure of Algorithm 5 should be skipped for certain small values of d (in addition to $d = 2$). However, the exact range of applicability of Algorithm 5 (that is, the minimum d from which it is safe to use this algorithm) has not yet been experimentally or theoretically determined. Such a study would require initial systems larger than what we have experimented with.

3 XL_SGE with Row Deletion (XL_SGE-3)

XL_SGE-2 demonstrates the benefits of using partial monomial multiplication. Instead of blindly skipping certain multiplications, we can adopt a more intelligent strategy. We first carry out all monomial multiplications. Subsequently, by analyzing the column statistics of the expanded system, we mark some equations as less important than the others. We delete the less important equations from the system and then perform SGE before the next stage of multiplication. This variant, henceforth referred to as XL_SGE-3, has one potential advantage over XL_SGE-2. Now, we have a better control over the initial reduction in the system size in the sense that the degradation of the rank profile can be carefully handled.

Algorithm 6: XL_SGE with Row Deletion (XL_SGE-3)

---

1. Expand the initial system $\mathbb{A}$ up to degree $d = 2$ using XL to obtain a linearized system $\mathbb{A}'$. Make a copy of the linearized system $\mathbb{A}'$ as $\mathbb{B}$.
2. Apply structured Gaussian elimination (SGE) with avalanche-control parameter K on $\mathbb{A}'$ to obtain a reduced system of equations $\mathbb{A}''$ of degree d.
3. Multiply the reduced system $\mathbb{A}''$ with monomials of degree 1 and linearize the system to obtain a system $\mathbb{A}'''$ of degree $d + 1$.
4. Identify and delete some rows of $\mathbb{A}'''$. Append the equations of $\mathbb{A}'''$ to $\mathbb{B}$. $\mathbb{B}$ now contains equations of degrees up to $d + 1$. Rename the system $\mathbb{A}'''$ as $\mathbb{A}'$.
5. If the degree of the system of equations $\mathbb{B}$ is D, end the process. Otherwise, go to step 2 after incrementing d by 1

Depending upon how we identify the redundant rows for deletion in Step 4, we have different variants of XL_SGE-3, some of which are elaborated below. The deletion of redundant equations can also be employed after Step 1 of Algorithm 6.

XL_SGE-3 with Deterministic Deletion Strategy (XL_SGE-3d)

We have considered only the variables of column weight two. Among the two equations containing a variable with column weight two, we delete (at most) one equation as follows.

Strategy 1

- If any of these two equations contains a variable with column weight one, then skip the deletion of both the equations. (In this case, the equation with the variable with column weight one is anyway deleted by SGE, thereby reducing the weight of the variable with column weight two.)
- Otherwise, delete the equation with the larger row weight. If both the equations have the same row weight, delete any one of these arbitrarily.

Strategy 2

- If any of these two equations contains a variable with column weight one, then skip the deletion of both the equations.

- If both the equations have the same right side (0 or 1), delete the equation with the larger row weight. Make arbitrary choices to break ties.
- If exactly one of the two equations has right side 1, then keep that equation, and delete the other.

Strategy 3

- If any one of the equations contains a variable with column weight one, determine whether that variable can reappear in the system in a future monomial-multiplication stage. If not, none of the equations is deleted. Otherwise, delete the equation containing the variable with column weight one.
- If both the equations contain variables of column weight one that can reappear from a future monomial-multiplication stage, then delete one of them depending on their row weights (as in Strategy 1).
- If both the equations contain no variables of column weight one, then take decision as in Strategy 1.

Let $z = x_1x_2x_3$ be a monomial with column weight one, and let the equation containing z also contain a variable with column weight two. In Strategy 3, we check whether z can reappear in the next multiplication layer (like multiplication of x_1x_3 by x_2). If that is the case, the current rank degradation incurred by the deletion of the equation containing z will be repaired later.

XL_SGE-3 with Random Deletion Strategy (XL_SGE-3r)

Let z be a variable (monomial) with weight t. We delete m of the t equations in which z appears. If the system is overdefined, this deletion is not expected to have a bad effect on the rank profile. The details of this strategy are given below. In our experiments, we have worked with $t = 2$ and 3, and $m = 1$.

- Find an equation with a variable of column weight t.
- If the equation contains a variable of column weight one, skip the deletion.
- Otherwise, delete the equation with probability p_d.
- Repeat this process until there are no removable equations with variables of column weight t.

4 Experimental Results

We have experimented with several variants of XL_SGE on small random systems (Table 1), and also on the initial system of size 890×208 obtained from four-round baby-Rijndael (Table 2). XL_SGE-2 and XL_SGE-3 significantly improves the performance of XL and XL_SGE.

5 Conclusion

The chief technical contribution of this paper is our efforts to improve upon the XL family of algebraic attacks. We suggest variants of XL_SGE. Our experiments establish the effectiveness of using our modifications in tandem with XL_SGE. Our proposals address some of the open problems of XL_SGE, but some other issues continue to remain unattended. Most importantly, a theoretical analysis of the XL_SGE family is needed. Here, we state some new avenues for research, that this paper opens up.

- The domains of applicability of XL_SGE$'$ need to be experimentally or theoretically determined.
- The dependence of the system size and rank profile on the seed (multiplication/deletion decisions) for XL_SGE-2 and XL_SGE-3r should be studied.
- An optimal choice for p (in XL_SGE-2) and p_d (in XL_SGE-3r) requires more experimentation and theoretical analysis.

Table 1: Performances of XL and variants of XL_SGE for random systems

Size of $\mathbb{A}$	Size of $\mathbb{B}$				
	XL	XL_SGE	XL_SGE-2	XL_SGE-3d	XL_SGE-3r
15×10	2712×637	2528×619	1447×631	1939×637	1360×637
16×11	2846×561	2119×561	943×561	1322×560	934×561
17×12	749×298	748×298	460×298	714×298	394×298
18×14	5347×1470	4796×1469	2199×1461	4356×1469	2462×1469
19×14	4831×1470	3620×1470	2333×1468	3447×1470	2414×1470
20×15	3783×1940	3963×1940	2907×1940	3149×1940	3073×1940
20×16	6402×2516	6094×2516	3700×2514	5407×2516	3994×2516
23×18	117996×31179	122701×31175	86200×31175	112307×31172	85227×31179

Table 2: Performances of XL and variants of XL_SGE for four-round baby-Rijndael ($D = 3$).

Algorithm	K	p	p_d	Size of $\mathbb{B}$	Rank Deficit δ
XL	0	1	0	2594060×1498713	96936
XL_SGE	3	1	0	2571848×1476481	93172
XL_SGE-2	0	0.75	0	2276971×1442363	89387
XL_SGE′	0	1	0	2556116×1449153	81576
XL_SGE-3d	0	1	0	1934149×1163740	79630
XL_SGE-3r	0	1	0.20	2355165×1449152	85470
XL_SGE-3r	0	1	0.25	2283125×1449152	89640

References

[1] J. C. Faugère, A new efficient algorithm for computing Gröbner basis (F4), 2000.

[2] J. C. Faugère, A new efficient algorithm for computing Gröbner basis without reduction to zero (F5), ISSAC 02, pp. 75–83, 2002.

[3] A. Kipnis and A. Shamir, Cryptanalysis of the HFE public key cryptosystem by relinearization, in CRYPTO, pp. 19–30, 1999.

[4] N. Courtois, A. Klimov, J. Patarin and A. Shamir, Efficient algorithms for solving overdefined systems of multivariate polynomial equations, in EUROCRYPT, pp. 392–407, 2000.

[5] N. Courtois and J. Pieprzyk, Cryptanalysis of block ciphers with overdefined systems of equations, in ASIACRYPT, pp. 267–287, 2002.

[6] J. Ding, J. Buchmann, M. Mohamed, W. Mohamed and R. Weinmann, MutantXL, in SCC, pp. 16–22, 2008.

[7] G. Bard, N. Courtois and C. Jefferson, Solution of sparse polynomial systems over GF(2) via sat-solvers, in ECRYPT workshop Tools for Cryptanalysis, 2007.

[8] S. Ghosh and A. Das, An improvement of linearization-based algebraic attacks, in InfoSecHiCom-Net, pp. 157–167, 2011.

[9] B. LaMacchia and A. Odlyzko, Solving large sparse linear systems over finite fields, in CRYPTO, pp. 109–133, 1991.