Some Algebraic Aspects of the Advanced Encryption Standard

Carlos Cid

Information Security Group,
Royal Holloway, University of London,
Egham, Surrey, TW20 0EX, U.K.
carlos.cid@rhul.ac.uk

Abstract. Since being officially selected as the new Advanced Encryption Standard (AES), Rijndael has continued to receive great attention and has had its security continuously evaluated by the cryptographic community.
Rijndael is a cipher with a simple, elegant and *highly* algebraic structure. Its selection as the AES has led to a growing interest in the study of algebraic properties of block ciphers, and in particular algebraic techniques that can be used in their cryptanalysis.
In these notes we will examine some algebraic aspects of the AES and consider a number of algebraic techniques that could be used in the analysis of the cipher. In particular, we will focus on the large, though surprisingly simple, systems of multivariate quadratic equations derived from the encryption operation, and consider some approaches that could be used when attempting to solve these systems.
These notes refer to an invited talk given at the Fourth Conference on the Advanced Encryption Standard (AES4) in May 2004, and are largely based on [4].

1 Introduction

Rijndael is a block-cipher with a simple and elegant structure. It has been designed to offer strong resistance against *known attacks*, in particular differential and linear cryptanalysis, while enabling efficient implementation on different platforms. Given its careful design criteria, it seems unlikely that its security can be affected by conventional methods of cryptanalysis.

Rijndael has also a highly algebraic structure: the cipher round transformations are based on simple operations over the finite field $\mathbb{F}_{2^8}$. Its selection as the AES has therefore led to a growing interest in the study of algebraic properties of block ciphers, as well as algebraic techniques that can be used in their cryptanalysis [1, 2, 6, 12, 13].

This new approach in cryptanalysis seems promising. One reason is that conventional methods of cryptanalysis of block-ciphers (e.g. differential and linear cryptanalysis) are generally based on a "statistical" approach: the attacker attempts to construct probabilistic characteristics through as many rounds of the

cipher as possible, in order to distinguish the cipher from a random permutation. Most modern ciphers have been designed with these attacks in mind, and therefore do not generally have their security affected by them.

In contrast, the so-called *algebraic attacks* exploit the intrinsic algebraic structure of a cipher: the attacker expresses the encryption transformation as a (large) set of multivariate polynomial equations, and subsequently attempts to solve such a system to recover the encryption key. Algebraic attacks could open new perspectives in the cryptanalysis of block ciphers.

In these notes we will examine some algebraic aspects of the AES and consider a number of algebraic techniques that could be used in the analysis of the cipher.

2 The Basic Structure of the AES

Rijndael is a key-iterated block cipher, which alternates key-independent round transformations and key addition. In the basic version (considered here), the cipher encrypts 128-bit blocks in 10 rounds, using 128-bit keys. We refer to [9] for a full description of the cipher.

In these notes we will also consider the *Big Encryption System (BES)*, another iterated block cipher which was introduced in [13]. BES operates on 128-byte blocks with 128-byte keys, and has also a very simple algebraic structure: one round of the cipher consists of inversion, matrix multiplication and key addition, all operations over $\mathbb{F}_{2^8}$.We refer to [13] for the full description of the BES cipher.

Both the AES and the BES use a *state vector* of bytes, which is transformed by the basic operations within a round. Furthermore, it is shown in [13] that one can *embed* the AES into the BES, and that way obtain an alternative description of the AES. This relationship between both ciphers may well provide new ways for the cryptanalysis of the AES.

3 Algebraic Analysis of the AES

Due to the rich algebraic structure of the AES, there is currently a growing interest in the study of algebraic techniques which could be applied in its cryptanalysis. These are known as *algebraic attacks*. Currently there appears to be two main approaches:

- Study the system of polynomial equations derived from the cipher;
- Study the AES underlying algebraic structure.

4 Algebraic Attacks

Unlike most conventional methods of cryptanalysis, the so-called *algebraic attacks* attempt to exploit the intrinsic algebraic structure of the cipher. More specifically, the attacker expresses the encryption transformation as a set of

multivariate polynomial equations and tries to recover the encryption key by solving the system.

While in theory most modern block ciphers can be fully described by a system of multivariate polynomial equations over a finite field, for the majority of the cases such systems prove to be just too complex for any practical purpose. However, given its algebraic structure, it seems that the AES could be more vulnerable to such approach.

In [6] Courtois and Pieprzyk exhibit a large, sparse and overdefined system of multivariate quadratic equations over $\mathbb{F}_2$ whose solution would recover the AES encryption key. In the same paper they propose a method called *XSL (eXtended Sparse Linearization)*, as an attempt to efficiently solve the system. Around the same time, Murphy and Robshaw [13] showed how to express the AES encryption as a far simpler system of equations over $\mathbb{F}_{2^8}$, which is derived from the BES. If XSL or some of its variants are in fact valid methods, this system should be faster to solve than the original one over $\mathbb{F}_2$, and in theory, could provide an efficient key-recovery attack.

5 Potential Attack Techniques

Given the AES and BES algebraic formulations, it is clear that an efficient method for the solution of this type of system of multivariate quadratic equations would provide a key-recovery attack of the AES with potentially very few plaintext-ciphertext pairs. While the problem of solving generic large systems of multivariate equations of degree greater than one over a finite field is known to be NP-complete, it is conceivable that a technique can be developed which exploits the particular algebraic structure of the AES and BES systems. Below we investigate a few approaches which have been proposed for solving such systems.

6 Linearization Methods

The method of *linearization* is a well-known technique for solving large systems of multivariate polynomial equations. In this method one considers all monomials in the system as independent variables and tries to solve the system using linear algebra techniques. In order to apply the method, the number of *linearly independent* equations in the system needs to be approximately the same as the number of terms in the system. When this is not the case, a number of techniques have been proposed to generate enough LI equations.

6.1 XL Algorithm

In [5] an algorithm for solving systems of multivariate quadratic equations called *XL* (standing for *eXtended Linearization*) is proposed. XL is a simple algorithm: if A is a system of m quadratic equations f_i in n variables over a field K, and $D \in \mathbb{N}$, one executes the following steps:

1. **Multiply:** Generate all the products $\prod_{j=1}^{k} x_{i_j} * f_i$ with $k \leq D-2$;
2. **Linearize:** Consider each monomial of degree $\leq D$ as a new variable and perform Gaussian elimination on the system obtained in step 1;
3. **Solve:** Assume that step 2 yields at least one univariate equation. Solve this equation;
4. **Repeat:** Simplify the equations and repeat the process to find the values of the other variables.

The hope is that after few iterations the algorithm will yield a solution for the system.

In [5] the authors present some estimates for the complexity of the algorithm for random systems with $m \approx n$. In particular, they provide evidence that XL can solve randomly generated *overdefined* systems of polynomial equations in subexponential time.

The XL algorithm (as in the form above) is a reasonably new idea, and its behaviour is not entirely understood yet. When analysing the algorithm, one must examine two key points:

1. Does the algorithm always terminate?
2. Does the algorithm work as predicted?

Does XL always terminate? By applying well-known commutative algebra techniques (Hilbert Theory), one can show that there are cases for which the algorithm does not terminate [10]. However, when working over finite fields, this problem can be avoided by adding to the system the underlying field equations $x_i^q - x_i = 0$.

Does XL work as predicted? Initially it was suggested that XL could solve systems of polynomial equations in subexponential time when the number of equations exceeded the number of variables by a small number. However there has been strong evidence that some of the heuristics used in the original article were too optimistic [3, 10].

This discrepancy arises from the fact that one may often overestimate the number of *linearly independent* equations generated by the algorithm. There has been recently few papers studying the XL [3, 10], and one could say that the algorithm has just started to be better understood now.

In any case, it is widely agreed that application of the XL algorithm against the polynomial system which arises from the AES (either over $\mathbb{F}_{2^8}$ or $\mathbb{F}_2$) does not provide an efficient attack against the cipher.

6.2 Variants of XL

Since the introduction of the XL method, a number of variants have been proposed. These attempt to exploit specific properties of the polynomial systems, such as how overdefined the system is, the order of the field, etc. Of particular relevance for the AES is the method proposed in [6] by Courtois and Pieprzyk.

XSL is based on the XL method, but uses the sparsity and specific structure of the equations to mount the attack; instead of multiplying the equations by all monomials up to certain degree, in the XSL algorithm the equations are multiplied only by "carefully selected monomials" (we refer to [6] and its earlier version [7] for a full description of the method). While this has the intention to create less new terms when generating new equations, it is not entirely clear the exact criteria used for selecting the monomials.

The system used in [6] to mount the attack has 8000 quadratic equations and 1600 variables, over $\mathbb{F}_2$ (the variables represent the input/output bits). Two attacks are described in [7]: the first one ignores the key schedule and therefore needs 11 known plaintext/ciphertext pairs (for the AES-128); the second attack uses the key schedule, and in theory could be mounted with a single known plaintext/ciphertext pair. In [6] it is claimed that the second XSL attack would have complexity of $\approx 2^{230}$ and $\approx 2^{255}$ when applied against the 128-bit and 256-bit AES, respectively. So the XSL attack would represent a (theoretical) successful key-recovery attack against the 256-bit AES.

XSL Attack on the BES. As said earlier, the $\mathbb{F}_{2^8}$-system derived from the BES is much simpler than the $\mathbb{F}_2$-system presented in [6]. In particular, it is far sparser. This would strongly suggest that the XSL attack is more suited to the BES system than to the original AES system.

Murphy and Robshaw consider in [13, 14] the consequences of the XSL attack against the BES. Using the estimates given in [6], they conclude that if XSL is in fact a *valid technique*, a key-recovery attack against the AES might be possible with a work effort of about 2^{100} encryptions. This would clearly represent a successful attack against the AES-128.

Accuracy of the XSL Estimates. The XSL algorithm consists basically of two main steps:

1. The equation generation procedure;
2. The T' method at the end of the algorithm.

The first step corresponds to the multiplication of the initial set of equations by selected monomials. This is done in similar manner of the XL algorithm. The T' method is used at the end, and in theory would allow the method to effectively solve the system even when the difference between the total number of terms and the number of linearly independent equations is reasonably large.

The main issue when considering XSL attacks (in fact, all the XL-based attacks) against the AES is how accurate the estimates for the number of *linearly independent* equations are. As explained above, there is evidence that some of the heuristics in the original XL paper were too optimistic. In fact, there is even more concern when considering the XSL algorithm. Additionally, it is not clear how effective the T' method is as a last step of the algorithm. The algorithm is an ad-hoc method, based on a number of heuristics arguments, and although this might not invalidate the XSL technique entirely, it makes it harder to formally

examine the algorithm and consider whether the XSL attacks described in [6] work *as claimed*.

In fact, we have considered very small versions of BES, with reduced block length and number of rounds, and smaller field. We ran a few simulations with these versions, and it appears that the attacks do not work in the manner predicted in [6]. Again, while this might not invalidate the XSL technique, it could raise doubts on whether the method is generally applicable against the AES and BES systems. It is clear that more research is needed to determine how effective this technique is against the AES.

7 Computational Algebra Techniques

Solving multivariate polynomial systems is a typical problem studied in Algebraic Geometry and Commutative Algebra. The classical algorithm for solving this type of problem is the Buchberger algorithm for calculating Gröbner Bases (see [8] for definitions and description of the algorithm). The algorithm generates a basis for the ideal derived from the set of equations, which can then be used to obtain the solutions.

The complexity of most algorithms used for calculating a Gröbner basis of an ideal is closely related to the total degree of the intermediate polynomials that are generated during the running of algorithm. In the worst case the Buchberger algorithm is known to run in double exponential time. One of the most efficient algorithms known, due to Faugère [11], appears to be single exponential. In any case, in practice it is widely believed that Gröbner Bases algorithms cannot be used for efficiently solving *generic* systems with more than a handful of variables.

However, the type of systems which arise from cryptosystems are often very structured. In particular, the BES system has a very regular structure. This is given for $j = 0, \ldots, 15$ and $m = 0, \ldots, 7$ by:

$$\begin{aligned}
&0 = w_{0,(j,m)} + p_{(j,m)} + k_{0,(j,m)}, && \\
&0 = x_{i,(j,m)} w_{i,(j,m)} + 1 && \text{for } i = 0, \ldots, 9, \\
&0 = w_{i,(j,m)} + k_{i,(j,m)} + \textstyle\sum_{(j',m')} \alpha_{(j,m),(j',m')} x_{i-1,(j',m')} && \text{for } i = 1, \ldots, 9, \\
&0 = c_{(j,m)} + k_{10,(j,m)} + \textstyle\sum_{(j',m')} \beta_{(j,m),(j',m')} x_{9,(j',m')}. &&
\end{aligned}$$

This system could be viewed as an "iterated" system of equations, with blocks of similar "sub-systems" repeated for every round. One could also use the transformation $\mathbf{x} \mapsto \mathbf{x}^{254}$ as the S-Box inversion to eliminate a number of variables (the BES system considered above has the simplest form, with only quadratic and linear equations).

Furthermore, since the system includes the equations relating every variable with its conjugates, we have the following easy proposition:

Proposition 1. *The degree of polynomials occurring in the computation of a Gröbner basis of a BES-type system with n variables is at most n.*

This is clearly an upper bound, and we expect that in practice the degrees are much lower. This fact, together with the particular structure of the system, can be exploited to infer more precise bounds for the complexity of the attack.

One can also seek alternatives for the use of the usual Gröbner bases algorithm. There are also a number of common techniques used in cryptanalysis that could be used in conjunction with computer algebra methods . For example, one could attempt to adapt the *meet-in-the-middle* technique and consider two smaller systems. This has the potential to reduce the complexity of the attack. One should also note that in practice the attacker is not primarily interested in the full solution of the system, but rather in the key variables. In fact, in a "partial key recovery" attack, only few key variables might suffice.

Therefore, it is possible that one may be able to use a combination of cryptanalytic and algebraic techniques (including Linearisation and Gröbner Bases) to mount a successful attack without actually computing the solution of the entire system.

7.1 The Polynomial Ideal Generated by the BES System.

Let S be the system of multivariate quadratic equations derived from the BES encryption operation, and K a fixed encryption key. A closer look at the properties of the ideal generated by these polynomials may prove to be useful when attempting to solve the system.

For every plaintext/ciphertext pair (P, C), we have a particular system $S_{(P,C)}$ and an ideal [1]

$$I_{(P,C)} = \langle S_{(P,C)} \rangle \subseteq \mathbb{K}[x_{i,(j,m)}, \ldots, w_{i,(j,m)}, \ldots, k_{i,(j,m)}].$$

In fact we are mostly interested in the ideal

$$I^K_{(P,C)} = I_{(P,C)} \cap \mathbb{K}[k_0, k_1, \ldots, k_{15}]$$

where $k_0, k_1, \ldots, k_{15}$ are the first key variables (i.e. the original key).

Thus for every key K, we can associate an ideal of $\mathbb{F}[k_0, k_1, \ldots, k_{15}]$ defined as

$$I_K = \bigoplus_{(P,C)} I^K_{(P,C)} \ ,$$

where (P, C) run through all plaintext/ciphertext pairs.

Given a key K, a random plaintext block P, and C such that $E_K(P) = C$, the probability that there exists another key K' with $E_{K'}(P) = C$ is approximately $(1 - 1/(e - 1)) \cong 42\%$. Therefore we expect that in many cases, for a given plaintext/ciphertext pair (P, C), the $\mathbb{K}$-dimension of the residue class ring $\mathbb{K}[k_0, k_1, \ldots, k_{15}]/I^K_{(P,C)}$ is greater than 1 (i.e., the corresponding reduced Gröbner basis should contain polynomials with degree greater than 1).

On the other hand, the $\mathbb{K}$-dimension of $\mathbb{K}[k_0, k_1, \ldots, k_{15}]/I_K$ is almost certainly 1. In other words, we expect I_K to be of the form

$$I_K =< k_0 - \kappa_0, k_1 - \kappa_1, \ldots, k_{15} - \kappa_{15} >$$

[1] To avoid inconsistent systems, we will make sure to describe the system in such way that it does not include "0-inversions" (i.e. use the map $x \mapsto x^{254}$ when necessary).

with $\kappa_i \in \mathbb{K}$. If this is not true, then there are *at least* two keys K_1 and K_2 such that

$$E_{K_1}(P) = E_{K_2}(P)$$

for *every plaintext block* P, and K_1 and K_2 induce the same permutation on the set of possible plaintext blocks, which would not appear to be the case for the AES.

8 Alternative Approaches

It is clear that an efficient method for solving the polynomial systems considered so far would represent a successful *key-recovery* attack against the AES. However, even when the system cannot be solved, other approaches could well be used in order to mount less ambitious attacks against the cipher. One could examine common applications of the AES, such as AES-based hash function and MAC constructions, modes of operation, relation between plaintexts, etc.

At the very least, a cryptanalyst would like to find a polynomial-time distinguisher between the cipher and a random permutation. This could be used either to mount a practical attack or simply to show some structural weakness of the cipher.

Given the rich algebraic structure of the cipher, it is not inconceivable that an "algebraic" distinguisher exists. This would most likely exploit the byte-oriented structure of the cipher and the typical round version of the BES, which consists of inversion, matrix multiplication and key addition over $\mathbb{F}_{2^8}$:

$$\mathbf{b} \mapsto M_B.\mathbf{b}^{-1} + (\mathbf{k}_B)_i$$

Mathematically, this seems to be the most natural representation of the cipher. Both the S-Box (inversion on $\mathbb{F}_{2^8}$) and the linear layer are highly structured, and this could well be exploited in the analysis of the cipher.

9 Conclusion

Rijndael is a cipher with a simple, elegant and *highly* algebraic structure. Its selection as the AES has led to a growing interest in the study of algebraic properties of block ciphers, with a particular focus on algebraic techniques that can be used in their cryptanalysis. One promising approach is to exploit the large, though surprisingly simple, system of multivariate quadratic equations derived from the cipher. An efficient method for solving this system would represent a successful key-recovery attack against the AES. While the problem of solving such systems is known to be hard, it is not entirely unlikely that a technique can be developed which exploits the particular algebraic structure of these particular systems.

Furthermore, it is also possible that the AES algebraic structure could be exploited on mounting less ambitious attacks. The AES has a rich algebraic

structure, and while many of these properties might not prove to be relevant in the cryptanalysis, it is not inconceivable that one could find a novel way to explore this structure in the analysis of the cipher.

References

1. Elad Barkan and Eli Biham. In how many ways can you write Rijndael? In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 160–175. Springer, 2002.
2. Alex Biryukov and Christophe De Canniere. Block Ciphers and Systems of Quadratic Equations. In *FSE'2003*, 2003.
3. Jiun-Ming Chen and Bo-Yin Yang. Theoretical Analysis of XL over Small Fields. In *Proceedings of the 9th Australasian Conference on Information Security and Privacy*, 2004. to appear.
4. Carlos Cid, Sean Murphy, and Matthew Robshaw. Computational and Algebraic Aspects of the Advanced Encryption Standard. In *Proceedings of the Seventh International Workshop on Computer Algebra in Scientific Computing - CASC 2004*, 2004. to appear.
5. Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations. In *Eurocrypt'2000*, pages 392–407. Springer, 2000.
6. Nicolas Courtois and Josef Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer, 2002.
7. Nicolas Courtois and Josef Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Cryptology ePrint Archive, Report 2002/044, 2002.
8. David Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms*. Undergraduate Texts in Mathematics. Springer, Second edition, 1997.
9. Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer-Verlag, 2002.
10. Claus Diem. The XL-algorithm and a conjecture from commutative algebra, 2004. submitted.
11. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner Bases without reduction to zero F5. In T. Mora, editor, *International Symposium on Symbolic and Algebraic Computation - ISSAC 2002*, pages 75–83, July 2002.
12. N. Ferguson, R. Shroeppel, and D. Whiting. A simple algebraic representation of Rijndael. In *Proceedings of Selected Areas in Cryptography*, pages 103–111. Springer-Verlag, 2001.
13. Sean Murphy and Matthew Robshaw. Essential Algebraic Structure within the AES. In M. Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 1–16. Springer-Verlag, 2002.
14. Sean Murphy and Matthew Robshaw. Comments on the Security of the AES and the XSL Technique. *Electronic Letters*, 39:26–38, 2003.