General Principles of Algebraic Attacks and New Design Criteria for Cipher Components*

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

Abstract. This paper is about the design of multivariate public key schemes, as well as block and stream ciphers, in relation to recent attacks that exploit various types of multivariate algebraic relations. We survey these attacks focusing on their common fundamental principles and on how to avoid them. From this we derive new very general design criteria, applicable for very different cryptographic components. These amount to avoiding (if possible) the existence of, in some sense "too simple" algebraic relations. Though many ciphers that do not satisfy this new paradigm probably still remain secure, the design of ciphers will never be the same again.

Key Words: algebraic attacks, polynomial relations, multivariate equations, finite fields, design of cryptographic primitives, generalised linear cryptanalysis, multivariate public key encryption and signature schemes, HFE, Quartz, Sflash, stream ciphers, Boolean functions, combiners with memory, block ciphers, AES, Rijndael, Serpent, elimination methods, Gröbner bases.

1 Introduction

In this paper we consider a very ambitious question: how to design secure cryptosystems and in particular how to design secure ciphers ? Very little real answers do exist in this area. However it is possible to learn from our experience, and formulate some design criteria, resulting on the one hand, from some practical requirements on cryptographic systems, and on the other hand, from the known attacks. Doing so we are still not done, and this for two reasons. First of all, the recommandations do usually conflict with each other and are not obvious to balance. Moreover for both practical implementation criteria and security criteria, it is always hard to know and debatable to what extent exactly a system satisfies these. Nevertheless, the work on the design criteria is and always was an important and necessary area of research.

This paper is about an emergence of a new type of design criteria on various types of cryptographic primitives. It turns out that many recent attacks on public key signature and encryption schemes, block and stream ciphers (including AES) have a common denominator. This common feature is the exploitation (by various methods) of the existence of various types of algebraic relations that involve both the inputs and the outputs of some component. We will formulate the resulting design criteria on the respective components that will be very similar, if not identical.

* Work supported by the French Ministry of Research RNRT Project "X-CRYPT".

2 From Boolean Functions to Algebraic Relations

Most of the current cipher design paradigms can be seen in terms of looking for in some sense "good" Boolean functions / "good" vectorial functions (S-boxes) and avoiding "bad" ones. The outputs of cryptosystems (and their components) should simply not depend on their inputs in a way that is too simple. The definition of the word "simple" does naturally vary from one place to another. For example in the design of stream ciphers, there are many so called "non-linearity" criteria, dictated by some (not always really practical) attacks. Building ciphers with such components allows to make sure that many (from real to very theoretical) attacks will not work very well on these ciphers. For example, in [27] Golic explains the criteria on the Boolean functions that should be used in stream ciphers. Obviously these criteria, to some extent being necessary in the design of good ciphers, are by far insufficient and nothing guarantees that a cipher that made out of "good" components will be good itself (i.e. will be secure). Moreover, using such components is sometimes even perceived (if they are really very good) as a potential danger (special may mean dangerous). In particular, many recent attacks in different areas of cryptography do work in spite of using very good (sometimes optimal) components w.r.t. aforementioned criteria (for example highly non-linear components).

2.1 Interesting Special Case: AES S-box

AES (Rijndael) [21, 22] is precisely a good case to study in this respect. First, because its security is simply essential, and more importantly, because it pushes the (aforementioned) philosophy that culminates two decades of research in the design of modern ciphers to its limits. A general question is, whether it is possible (and how) to attack ciphers build with highly-nonlinear components (and thus build with eminently "good" Boolean function. Obviously studying this question will in most cases not give results being directly applicable to AES, but it gives us the opportunity to come up with new approaches to attack AES later, as well as should help us to simply design much better ciphers in the future (that avoid also the recent attacks).

In [6], Canteaut and Videau study the non-linearity properties of the Inverse function in $GF(2^n)$ (the only non-linear component of AES) with relation to linear, differential and higher-order differential attacks. It is exceptional and close to optimality, see [6]. On page 6 of [23], the designers of AES say: *"[...] The disadvantage of these boxes is that they have a simple description in $GF(2^m)$, [...] we are not aware of any vulnerability caused by this property. [...] Should such a vulnerability exist, one can always replace the Sboxes by Sboxes [...] that are not algebraic over $GF(2^m)$. [...]"* Unfortunately important vulnerability of the inverse S-box does exist. Historically the idea goes back to the algebraic attacks on several so called multivariate public key schemes, initiated by Patarin in [43], greatly improved by Courtois et al. [10, 20], and recently adopted by Faugère and Joux [33]. The seminal idea (due to Patarin) is to study the security of a cipher component not in terms of Boolean/algebraic functions, but in terms of Boolean/algebraic **relations** that involve both inputs and output bits. In the last two years, this precise idea, has led to a sudden collapse of several important

families of stream ciphers, as demonstrated by Courtois, Meier et al in [18, 19, 2, 11, 14] and numerous other recent papers. We explain these in Section 4. But does it matter at all for block ciphers ? This will the main subject of this paper starting from Section 5.

3 From Multivariate Public Key Schemes to General Algebraic Attacks

At Crypto'95, Jacques Patarin proposes a very interesting attack on the Matsumoto-Imai public-key cryptosystem of Eurocrypt'98, see [38, 42]. This cryptosystem, at the time considered as very promising, is based on a univariate transformation, that can be for example $X \mapsto X^3$. This cube function, instead of being over a ring of numbers modulo some N like with RSA, is over a finite field, for example $GF(2^{80})$. The order of a multiplicative group of $GF(2^{80})$ is known and therefore in many cases, such a power function over a finite field is, unlike in RSA, easily invertible. However, the same algebraic structure of this function can be "concealed" (cf. [38, 42]) when it is written in a new representation, as a set of multivariate quadratic polynomial functions. It is done in such a way that it is easy to compute it forwards, and hard backwards, for anyone that does not known how the system of equations have been generated. Thus, Matsumoto and Imai construct their public key cryptosystem, see [38] for more details.

Incidentally, due to the cube function, this cryptosystem have extremely good properties when considered in terms of Boolean functions, see [41, 6]. Yet, this did not prevent Jacques Patarin from rather badly breaking this cryptosystem, at Crypto'95 [42]. The attack is extremely interesting. He shows that there are simple algebraic relations that relate input and output bits of this cryptosystem. More precisely, if the input is $(x_0, \ldots, x_{79})$ and the output is $(y_0, \ldots, y_{79})$ there exist bi-linear equations of type, for example $\sum_{ij} \alpha_{ij} x_i y_j = 0$. Then, Patarin remarks that if such equations exist, they can be easily found from the public key, and then subsequently they can be used to decrypt any message: if we substitute a concrete values of y in these equations they become linear and can be solved to recover the x_i.

This attack has been generalised by Courtois in [10]. This paper also proposes a first "theory" of algebraic attacks on public key schemes [1] that we will develop and explain here. This "theory" is quite simple and can potentially be applied to many different situations that arise in cryptanalysis. To achieve this we will be voluntarily imprecise. Some details vary from one attack to another, and it should be applicable also to situations that are very different than the area of algebraic attacks.

From one point of view, one can think that it applies to more or less all cryptographic attacks. To explain this, let's consider any attack on any deterministic one-way function which is described as a set of explicit arithmetic formulae $y_i = F_i(x_0, \ldots, x_{n-1})$. The answer x we are looking for is also seen as a set of equations, though much simpler $x_i = \ldots$, which a hypothetical attack would

[1] It applies also almost literally to algebraic attacks on block and stream ciphers, but at the time, nobody really suspected this.

evaluate to. We wish to look at any deterministic attack as a series of transformations that starts from (somewhat "complex") initial equations and eventually produces somewhat "simpler" ones (containing the solution to the system). Similarly, following [10], starting from some notion of complexity that is adapted to our initial equations, and makes them hard to solve, we can also try to construct attacks that work exactly in this way. For this, still following [10], we need to study (and find) methods that given some initial equations, give hope to generate some "simpler" equations. With such methods we hope to solve the system, by successive simplifications. For example, one possible notion of complexity is the non-linear degree. In Matsumoto-Imai and HFE systems [38, 42, 43] we have initial equations that are quadratic and our goal will be to find some simpler, linear equations. Most attacks known on these systems work in this way, e.g. [42, 43, 10, 20, 33].

Attacks that work in such a way, may be iterative with many steps, which makes them hard to understand and study. For example it is far from being clear what is the complexity of Courtois-Pieprzyk XSL attack on AES [17]. However, again following [10], what one should study, and what is really interesting, is what happens in one step of the attack. From the cryptological point of view the main question will be not what is the exact complexity of an attack, but rather if the attack is feasible in general (at least in some cases), and even more importantly, how to completely avoid such attacks. For these questions, the most important answers may already be given by looking at the beginning of the attack process. Do we gain something ? Can we by some means gain anything simpler from the initial equations ? Obviously it is always possible to combine equations in some way, (and it is very simple for Boolean algebraic equations over a finite filed). However, usually, we obtain other equations that have nothing special and are in fact more complex than the initial equations. Following [10], the interesting phenomenon to watch for is a type of "collapse in the complexity". For example, we take some multivariate equations of degree 2, combine them algebraically to get an equation of expected degree 4, but when we compute this equation its degree collapses from 4 to 3. Here we gain something, some simplification arises. The heuristic is then that, if it can be done once, it can be done several times and in many cases we end up by obtaining a full working algebraic attack. In rare cases, it will obviously fail, but we know that designing systems such that there is no "collapse of complexity" in some sense, will prevent many attacks, whether they work well, or not. For example, building a cipher with large random components(e.g. S-boxes) makes such cases of "complexity collapse" to some degree very unlikely if not impossible, this whatever is our definition of complexity.

When, as in many cases studied in this paper, the notion of complexity is the non-linear degree of a multivariate polynomial form of a function, the existence of "complexity collapse" can be characterised as follows. If an algebraic combination of the original equations is of lower degree than expected, it means that there exist a non-trivial and in some sense "simple" (e.g. low degree) function G such that:

$$G\left(x_0, \ldots, x_{n-1};\quad F_0(x_0, \ldots, x_{n-1}), \ldots F_{m-1}(x_0, \ldots, x_{n-1})\right) = 0$$

If we replace $y_i = F_i(x_0, \ldots, x_{n-1})$ we get an algebraic relation between input and output bits:

$$G\,(x_0, \ldots, x_{n-1};\quad y_0, \ldots, y_{m-1}) = 0 \qquad (*)$$

In these formulas the x_i and the y_i may be in $GF(2)$, but may be also in any other finite field $GF(q)$. We are at the right point. It turns out that talking about algebraic relations is more general than considering "a collapse in the complexity": algebraic relations may exist, be found and directly be used in an attack, disregarding the initial complexity of the equations, that in some cases is within no comparison (much more complex).

Undoubtedly, there are many cases in which the very existence of an algebraic "complexity collapse" or/and resulting algebraic relations at some level, is somewhat trivial and inevitable. There are also many cases in which such occurrence can be an isolated phenomenon that does not lead to interesting attacks. Yet, to make sure that a system resists to large class of possible attacks it is sensible to avoid such situations whatsoever. (This concerns, as we will explain later mainly Generalised Linear Cryptanalysis and direct algebraic XSL-type attacks, and potentially other future attacks). Another way of seeing such design criteria is to say that, in a sense, components of our system (or the whole system itself) will be "more" indistinguishable from random components (e.g. random functions or random permutations), and thus less attacks should be possible.

In the following sections we will explain briefly, how this general paradigm of algebraic attacks applies to other contexts. This list is not exhaustive, and we expect that many other areas of cryptographic security can be described in a meaningful way in terms of "complexity collapse" and/or simple "I/O relations" with respect to some (not necessarily algebraic/polynomial degree) notion of complexity.

3.1 How to Build Secure Multivariate Public Key Cryptosystems

Here the conclusion follows immediately: for a trapdoor function to be secure we need to make sure that there is no multivariate relations such as $(*)$ that contain less than, let's say 2^{80} different monomials (in general, for finite systems, it is impossible to avoid the existence of algebraic relations, but their size will be astronomical). In practice, for most systems, if there is no algebraic/multivariate relations of size less than 2^{40}, there should be no practical algebraic attack on the system (because we need to be able to recover the equations first). However, in some special cases, equations of large sizes can be build directly by a method that depends on the cipher, and then they can be used by substitution of variables. Therefore the proposed bound of 2^{80} gives a better guarantee.

4 Algebraic Attacks on Stream Ciphers

The algebraic attacks on stream ciphers have been introduced in 2003 by Courtois and Meier [19, 18]. Since then, the area has known an important research activity with many interesting contributions, to quote only some, by Armknecht and Krause [2, 1], Cho and Pieprzyk [7], Courtois [14, 11], Hawkes and Rose [29], Lee, Kim, Hong, Han and Moon [35], Meier, Carlet and Pasalic [39], and others. In this paper we only explain the main principle of algebraic attacks on stream

ciphers from [19, 11], and what are the resulting design criteria for components of such ciphers.

The algebraic attack on stream ciphers is extremely general and applies potentially to all ciphers that have some linear feedback (for example based on LFSRs or cellular automata). We assume that in our cipher the first (linear) component is as follows. Let $x = (x_0, \ldots, x_{n-1}) \in GF(q)^n$ be the state of this component. We assume that the cipher is regularly clocked (some relaxations are possible, see [19, 18]) and at each clock the linear state x is updated by some multivariate linear function L. This means that at each clock x becomes $L(x)$, and if $K = (K_0, \ldots, K_{n-1})$ is the initial state, at time t the state will be called $x^{(t)}$ and by definition we have $x^{(t)} = L^t(K)$.

Then we assume that the state of the linear component is supplied to the second "filter/combiner" component that outputs the keystream (it may output one or several bits at a time). This output component can be stateless or stateful: in the second case it also has internal memory bits that are updated at each clock. In this case, we have in addition to the linear feedback in the first component, a non-linear feedback in the second component (but usually of much smaller size/importance than the linear feedback).

Let l be the number of memory bits in the second component, that before and at the time t are $a_0^{(t-1)}, \ldots, a_{l-1}^{(t-1)}$. In particular, for stateless filters/combiners l is 0, for example when a Boolean function is used to filter/combine the state bits of one or several LFSRs. The initial inner state is $a^{(-1)}$, exists before $t = 0$, and can be anything (it is unknown in the attack and algebraic attacks tend to eliminate all the monomials in the a_i). At each clock $t = 0, 1, 2, \ldots$, the combiner outputs m bits $y_0^{(t)}, \ldots, y_{m-1}^{(t)}$, for $t = 0, 1, 2, \ldots$. For example, if the ciphers uses a single Boolean function to combine input bits, we have simply $m = 1$. In general, the second component can be described as a pair of functions $F = (F_1, F_2) : GF(2)^{n+l} \rightarrow GF(2)^{m+l}$, that given the current state and the input, compute the next state and the output:

$$F : \begin{cases} (y_0^{(t)}, \ldots, y_{m-1}^{(t)}) = F_1(x_0^{(t)}, \ldots, x_{n-1}^{(t)}, a_0^{(t-1)}, \ldots, a_{l-1}^{(t-1)}) \\ (a_0^{(t)}, \ldots, a_{l-1}^{(t)}) = F_2(x_0^{(t)}, \ldots, x_{n-1}^{(t)}, a_0^{(t-1)}, \ldots, a_{l-1}^{(t-1)}) \end{cases}$$

The most general form of an algebraic attack on stream ciphers following closely [11, 14, 19] works as follows.

- We assume that L is known (for example the LFSRs used in the cipher are known or can be guessed/revovered).
- We consider M consecutive states of the cipher.
- Find (by some method that is very different for each cipher) one (at least, but one is enough) multivariate relation G between the state bits x_i and some M consecutive outputs, for example:

  $$G(x_0, x_1, \ldots, x_{n-1};\ y^{(0)}, \ldots, y^{(M-1)}) = 0$$

  We assume that G is of degree d in the x_i (the degree in the y_i may also be important, but usually will not influence the total attack complexity).
- By recursive structure of the cipher, for any initial state K and for any t, the same equation will apply to all consecutive windows of M states

$$G(L^t(K);\ y^{(t)}, \ldots, y^{(t+M-1)}) = 0$$

- The $y^{(t)}, \ldots, y^{(t+M-1)}$ are replaced by their values known from the observed output of the cipher.
- Due to the linearity of L, for any t, the degree of these equations is still d.
- For each keystream bit, we get a multivariate equation of degree k in the x_i.
- Given many keystream bits, we inevitably obtain a very overdefined system of equations (i.e. great many multivariate equations of degree d in the K_i).
- To solve these equations we may apply the XL algorithm from Eurocrypt 2000 [13], adapted for this purpose in [18] and other improved elimination techniques such as computing Gröbner bases combined with linear algebra, see [24, 25]. However, if we dispose of a sufficient amount of keystream, (which is frequently not very big, see [19]), all these are not necessary.
- If the amount of keystream available is large enough, we use a so called linearization method that is particularly simple. There are about $T \approx \binom{n}{d}$ monomials of degree $\leq d$ in the n variables K_i (assuming $d \leq n/2$). We consider each of these monomials as a new variable V_j. Given about $\binom{n}{d} + M$ keystream bits, and therefore $R = \binom{n}{d}$ equations on successive windows of M bits, we get a system of $R \geq T$ linear equations with $T = \binom{n}{d}$ variables V_i that can be easily solved by Gaussian elimination on a linear system of size T. The time to solve such a system is T^ω with in theory $\omega \leq 2.376$ [8] but in practice for small systems, it is believed that one should rather consider ω that is closer to 3 than to 2.376.

4.1 How to Build Secure Stream Ciphers

For stream ciphers in which the second component does not have internal memory, the case $M > 1$ does not make a lot of sense, and if we wish the cipher to avoid algebraic attacks, we get a requirement on the second component that is identical to our requirement on public key trapdoor functions. There should be no "simple" algebraic relations between its inputs and outputs such as:

$$G\,(x_0, \ldots, x_{k-1};\ \ y_0, \ldots, y_{m-1}) = 0 \quad (*)$$

Similarly, in the general case $l \geq 1$ we need to avoid the existence of "not too complex" equations (that eliminate the internal state bits a_i) of type:

$$G(x_0, x_1, \ldots, x_{n-1},\ y^{(0)}, \ldots, y^{(M-1)}) = 0 \quad (**)$$

For stream ciphers however, the notion of "simple" and "complex" equations changes. It is no longer the total size of these equations (number of monomials) that matters, but their degree in the x_i (their degree in the y_i can be large, provided that the total size of the equations is not too big and that there is some method to generate these equations from the description of the cipher). Our recommandation, for ciphers that aim at 2^{128} security is that there should be no G that can be **efficiently written** (for example using up to 2^{128} of memory) with degree $d \leq 16$. (We do not exactly require that they do not exist, and for some high d there may exist large relations with, for example 2^{100} monomials, that are not a problem as long as there is no efficient algorithm to recover/write

and otherwise use them). For higher security levels, for example military-level requirements of type 2^{256}, we recommend a cautious $d \geq 32$. For specific ciphers these numbers may be lower but then they require a careful study if they will not be broken by fast algebraic attacks [14, 1, 29].

It is certainly possible to obtain components that satisfy these criteria by using sufficiently large random S-boxes (the exact size will depend a lot of the exact construction). Otherwise, proposing constructive methods to obtain components that will (if possible provably) satisfy these criteria is an important open problem. For Boolean functions, this problem can be rephrased as constructing "good" Boolean function that in addition to classical non-linearity criteria respond also to the new criterion of "algebraic immunity". It also remains an open problem, see [39].

5 Block Ciphers and Algebraic Relations

This paper is about a simple idea of studying algebraic relations on different components. In this paper we will not try to summarise all the results but the outcome of this approach on stream ciphers and multivariate public key schemes was quite devastating, see among others [1, 2, 7, 10–12, 14, 16–20, 24, 25, 29, 33, 35, 39, 42–44, 13]. Several classes of schemes were shown to be substantially less secure than expected, and sometimes badly broken. But the real question that many people are asking is, does this type of attacks matter also for block ciphers ?

At present many cryptologists still believe that they don't matter (at all). Yet, from one point of view there is no doubt that it does ! For example with the polynomial approximation attack of [32], Jakobsen was the first to claim that to obtain secure ciphers *"[...] it is not enough that round functions have high Boolean complexity. [...]"* . He proposes already to avoid functions that have simple algebraic properties in the design of block ciphers (but his warning was never taken seriously). Regarding the AES S-box, in [9] and in [15] in these proceedings, Courtois shows that it is possible to construct, by several very different methods, many block ciphers based on the inverse in $GF(2^n)$ that satisfy all the known design criteria on block ciphers, yet remain very very weak.

These schemes are insecure, because the Inverse-based Rijndael-type S-boxes, though very complex when regarded as a function, can be characterised in several ways by algebraic relations, cf. [15, 17, 40]. Here we are concerned with attacks being forms of generalised linear cryptanalysis, see [28, 34, 9, 15]. Though these attacks techniques clearly do evolve into general attacks that can be applied potentially to any block cipher, the insecure ciphers constructed in [15] remain very special contrived ciphers.

On the contrary, for ciphers such as DES and AES, that use relatively small S-boxes and a lot of diffusion that connects the outputs of one S-box to many other S-boxes in the next rounds, (wide trail strategy of AES designers [21, 22]), we expect that the things should be very different. In [15, 9], heuristic arguments are given to the effect that, the impact of generalised linear cryptanalysis on such ciphers (e.g. AES) is expected to be low, as long as they resist well to linear

cryptanalysis. Therefore, it seems so far that the algebraic relations may do not really matter so much for AES and similar ciphers.

6 Global Algebraic Attacks on Block Ciphers

We see that, finding attacks on ciphers such as AES, remains an ambitious task, even given the existence of algebraic relations on the S-boxes. Unfortunately, there is yet another attack strategy, published in 2002 by Courtois and Pieprzyk, that is designed to render the "wide trail strategy" useless. It can be called a **direct/global** algebraic attack strategy, or **exact** algebraic approach. At the origin, it also uses the existence of algebraic relations for the individual components of the cipher. We do not however try to connect the specific monomials that appear in one equation to another equation, which may be very hard, but just write the equations for the whole cipher, to obtain a global system of equations that uniquely characterizes the key to be found. Then we see if it is possible (in theory and/or in practice) to solve such a system of equations.

This type of approach, if proven to work efficiently in practice, is not less than a major revolution in the field of block cipher cryptanalysis. This is because, except few very weak ciphers, all the general attacks known up till now for block ciphers are attacks that combine "approximations", that are some properties (linear, differential, higher-order differential, polynomial approximation etc..) true with some probability that except for some very weak ciphers is different from 1. This "combine approximations" paradigm has three important consequences. First of all, the complexity of the attacks must grow exponentially with the number of rounds. Secondly, the number of plaintexts needed in an attacks also grows in the same way (and may be the main limitation in practice). Finally, ciphers with good diffusion (wide trail strategy) force the attacker to use several approximations in parallel in the same round, and the efficiency of the attacks further decreases.

The "exact algebraic" approach that exploits equations that are true with probability 1 that exist locally (for example for each S-box) has the potential to remove simultaneously the three aforementioned obstacles. The complexity is not longer condemned to grow exponentially with the number of rounds. The number of required plaintexts may be quite small (e.g. 1). And the wide trail strategy should have no impact whatsoever on the complexity of the attack.

6.1 How Secure are Today's Block Ciphers ?

Some people dismiss the idea of an algebraic attack on AES, as being too simple and too naive to be true. Our impression is that, it is rather the current thinking about the security of block ciphers that is very naive.

We get the impression that, if we mix sufficiently many rounds of any construction, it will be secure. In practice however, the ciphers are meant to be rather fast, have a limited number of rounds, but yet the security claims made on them are extremely ambitious. During the AES contest many authors proposed ciphers claimed to be indistinguishable from a random permutation within less than 2^{256} computations. This is a huge number. With the Moore's law, such keys should remain secure against brute force until around 2200. This gives us

200 years to invent new mathematics, new algorithms, and new attacks that will break the cipher faster than the exhaustive search before it is outdated. Who can make security predictions for such a long period of time, knowing that so many security claims are disproved every year ? Moreover, 2^{256} is close to the number of atoms in the universe, therefore it also possible that the computers will never actually have such a computing power. This means that we are left with infinite time to find better attacks. We believe therefore that betting that a cipher cannot be distinguished from a random cipher faster than by brute force, may be an infinitely risky bet for 256-bit ciphers. Our guess is rather that **all** the block ciphers with 256-bit keys that were submitted to AES, will some day be broken faster than by exhaustive search, simply because our current knowledge about the real security of block ciphers is yet very low.

6.2 Who Invented Algebraic Attacks on Block Ciphers ?

According to a visionary recommandation of Shannon from his 1949 paper [45], breaking a good cipher should require: *"as much work as solving a system of simultaneous equations in a large number of unknowns of a complex type"*. There are many ways of interpreting this statement. For example we may think about multivariate quadratic equations with Boolean variables, the large number of unknowns may mean a large number of monomials, unknowns of a complex type may mean monomials of high degree (or that combine variables that come from remote locations inside a cipher).

From another point of view, it is a trivial folklore attack that anyone can think of. Indeed, it is easy to see that, for any practical cryptographic system that relies on computational (not information-theoretic) security, we can write a system of Boolean equations such that solving it allows to find the key. Then, solving a general system of Boolean equations is an NP-hard problem, and solving non-linear systems of large size is expected to be hopeless. However, it turns out that, what makes such problems hard is not so much the number of variables or monomials, but the balance between the number of equations and the number of monomials that appear in these equations. From this, we expect that, systems that are overdefined, sparse, or both, should be much easier to solve than general systems of similar size. As far as we know, before 1998-2000, the scientific community were not aware of this fact, and easily believed that large systems of equations are necessarily hard to solve. When the XL attack was first introduced by Courtois, Klimov, Patarin and Shamir [13], as a development of earlier ideas of Shamir and Kipnis [44], things started to change. In particular, specialists of elimination methods such as Gröbner bases that have been studied for many years now, see for example [46, 24, 25], started to realise the full potential of these and other algebraic techniques to solve problems that arise in cryptography. It turns out that the cryptographic instances of multivariate systems of equations have several interesting properties that may and do help to solve them efficiently. Among these properties we will quote the fact that they are over very small finite fields, they usually have a unique solution, they do not have solutions in extensions fields or at infinity, and again, they are frequently over-determined, and sparse (with several possible notions of sparsity).

At present the area of algebraic attacks is full of open problems that should be solved with time. A lot remains to be done in discovering cryptanalytic applications of already existing algebraic methods of solving systems of polynomial equations. Similarly, specific systems of equations that arise in cryptography should allow (and already do) to better understand why certain very general algebraic algorithms (such as Buchberger or F5 algorithms) for solving equations work well in some cases, and do fail in some other cases. Finally, new methods of solving algebraic equations should and will be invented, motivated by cryptographic attacks.

6.3 The Structure of Algebraic Attacks

Global algebraic attacks on block cipher following Courtois and Pieprzyk (previously imagined also by Shannon, Patarin and probably few others) contains the following three stages, that can (and probably should) be studied separately.

1. **Write an appropriate initial system.** Write a system of equations that, given one or several known plaintexts, uniquely characterizes the key. This system should be as over-determined (also called overdefined) and as sparse as possible. This can be measured by the initial ratio R_{ini}/T_{ini} between the number of equations R_{ini} in the system and the total number of monomials T_{ini} that appear in it. It can be for example $1/4$ or $1/3$. It is not clear what is the optimal setting for algebraic attacks: we may try simply to achieve a lowest R_{ini}/T_{ini} possible, however for some systems with a higher initial ratio, but a lower global size, or some specific additional properties, the overall complexity of an algebraic attack may be lower.
2. **Expand it.** The second step is an expansion step. The goal is, starting from the original R_{ini} equations with T_{ini} monomials, to produce (for example by multiplying the equations by some well chosen polynomials) another (much bigger) set of R equations with T monomials. The goal is to have the new ratio R/T close (or bigger than) 1. If $R > T$ it means that the set of equations is redundant, and we should think of a better method of generating them (to avoid redundancies) and also of a better method of counting how many equations we have, that are not trivial linear combinations of other equations, and therefore serve no purpose.
   Here the main criterion of "success" is not so much the final ratio R/T (that simply must be somewhat close to 1, e.g. 0.9) but the size T. However it remains possible that some attacks with a worse (larger) T and better (bigger) R/T do in fact work better (cf. next step).
3. **Final in place elimination.** The final step should be an "in place" elimination method that given an "almost saturated system" with R/T close to 1, finds a solution. On proposed method to achieve this is by generating a completely saturated system (the T' method proposed by Courtois in [17, 16]. It can also be achieved by computing a Gröbner basis of the expanded system, and probably by other means. The (heuristic) requirement is that the memory required in this third step should not exceed T, otherwise maybe we need to improve rather the second (expansion) step.

6.4 Applicability of Algebraic Attacks

There are reasons why, overdefined and/or sparse systems are bound to appear frequently in cryptography. In most settings, there is no cryptographic solutions with unconditional security, and we have to rely on computational security. A relatively short (128 bits or less) key will be usually used many times, to produce much more information: many known plaintexts, many signatures, etc. In public key cryptography, a proof of security would allow to be certain that each utilization of the cryptographic scheme, does not leak useful information. Secret key schemes do not have such proofs of security, and the more we use it, the more the problem become overdefined (if we do not introduce additional variables). It is also in secret key cryptography, that the problems may become really massively overdefined, if we think about the amounts of data that can be encrypted with a single key, on a satellite link. Another problem is a consequence of the fact that many ciphers are designed to be implemented in hardware with a very low gate count. This allows to design an algebraic attack with relatively small umber of variables and a very small number of monomials (very sparse).

These are theoretical considerations. The present experience of algebraic attacks is that, their complexity should grow "nearly polynomially" in the number of rounds and in the block size, with however a really huge constant called Γ that does depend only on the S-box. (This for all known versions of the XSL attack, and for both resulting definitions of Γ, see [17]). For a random S-box (and also for many other S-boxes that have no special properties such as algebraic relations) this constant Γ can be shown to be double-exponential in s, the size of the S-box in bits. In [17], it appears that already 4-bit S-boxes, should be sufficient for 2^{128} security and probably beyond. For the Rijndael S-boxes, it is possible to see that Γ grows only simply exponentially in s. Then it seems that even for $s = 8$ algebraic attacks faster than 2^{128} may exist, see [17, 40], but we are clearly on the frontier of applicability of algebraic attacks. Thus, it seems that in fact algebraic attacks are only possible for some very special ciphers. Apart from Serpent and Rijndael, we are not aware of a single other block cipher for which even a current (probably too optimistic) estimations of the complexity of algebraic attacks would give less than the exhaustive search.

6.5 Is AES Broken ?

It is important to say: we really do not know. It is possible that, one of the XSL attacks works quite well, or a simple combination of already known attacks already breaks AES. Our favorite candidate in this respect would be to combine the Murphy-Robshaw idea of using equations over $GF(256)$ from [40], with one of the XSL expansion attacks from [17], and replace the final T' method by a (presumably better) advanced Gröbner bases algorithm such as Faugère's F5 [25]. This might simply break AES. But it is also possible that it fails quite miserably for some fundamental reason that is not yet understood. Then, a slight modification of the attack could still remove the theoretical obstacle and give an attack that might again work in practice. Studying algebraic attacks on block ciphers in all due details is outside the scope of this paper, and remains

largely to be done. Both theoretical and experimental results will probably be needed to get the full picture.

6.6 How to Avoid Algebraic Attacks on Block Ciphers

At any rate, we advocate to take the algebraic attacks on block ciphers very seriously and to design block ciphers that do avoid such attacks. The resulting security criterion is, still more or less the same. The S-boxes of a block cipher should avoid the existence of "simple" algebraic relations of type:

$$G(x_0, \ldots, x_{s-1};\ \ y_0, \ldots, y_{s-1}) = 0 \quad (*)$$

The exact definition of "simple" that would prevent all algebraic attacks on block ciphers is not obvious to give. We need to avoid equations that, for some representation, and some system of equations, give a low value of Γ. For example following [17], we should avoid systems that are too overdefined or/and too sparse.

This should not be very hard to achieve. We believe that using random S-boxes on 8 bits should be about sufficient to achieve 128-bit security (though not for sure). We recommend in fact to construct bigger S-boxes that have no algebraic relations starting from random bijective 8-bit S-boxes. For higher security requirements such as military applications, we advocate to make mandatory a requirement that the cipher should use at several places inside the encryption, a random S-box of at least 16 bits.

7 The Future of AES

In our opinion, AES should still be recommended as the best choice of encryption algorithm for applications that do not require long-term security. We believe however that NIST should set an expiration date for AES, that could be 2010. It could be extended it later, according to the developments in cryptanalysis, but we believe that in 2010 it will be much wiser to replace AES by a better cipher, being not vulnerable to algebraic attacks, generalised linear cryptanalysis with multiple approximations, and other attacks that will probably be invented by 2010. The replacement should be done even if it turns out that known algebraic attacks on block ciphers do not work, and all other attacks that exploit algebraic relations (e.g. generalised linear cryptanalysis) do not break AES either.

In addition, we believe that a cipher such as AES can only be really credible as the world's standard all-purpose cryptographic high security lock, if there is a series of AES challenges. They could range from 100 to 1 million dollars, and be offered for solving various important open problems that in a different manner do compromise the security of AES, up to a total break that is done or doable in practice. This would allow to monitor the progress in the security of AES and to ascertain a very serious status of this scheme, compared to so many other schemes that are broken every year. For people that do not have expertise in cryptography, and cannot tell between real or fake security experts, such challenges, are **the only** way of knowing that the AES is indeed not yet broken, and also to see that some people take its security seriously enough to offer 1 million dollar to whoever demonstrates it can be broken in practice.

8 Conclusion

Algebraic attacks exploit the existence of multivariate relations on the appropriate cryptographic component. They do allow to break many multivariate public key schemes and stream ciphers. For block ciphers, their effectiveness is far from being clear. Yet, it is very sensible to avoid the existence of such algebraic relations for non-linear components of block ciphers. This not only because of algebraic attacks, but also because of generalised linear attacks: examples of contrived ciphers are known that are not secure with relation to these.

Thus, we propose (if possible) to simply avoid multivariate and algebraic relations in **all** types of cipher components. This extends the current paradigm of avoiding "bad" Boolean functions, or/and "bad" vectorial functions (S-boxes).

One method to achieve this would be to construct appropriate cryptographic components with guaranteed "algebraic immunity". A much simpler method, is to use sufficiently large random S-boxes. This should prevent all known attacks on block ciphers: linear/differential cryptanalysis with generalisations, all kinds of generalised linear attacks as described in [15], and also any kind of exact algebraic attacks such as XSL [17].

References

1. Frederik Armknecht: *Improving Fast Algebraic Attacks,* to appear in FSE 2004, LNCS, Springer.
2. Frederik Armknecht, Matthias Krause: *Algebraic Atacks on Combiners with Memory,* Crypto 2003, LNCS 2729, pp. 162-176, Springer.
3. Kazuaro Aoki and Serge Vaudenay: *On the Use of GF-Inversion as a Cryptographic Primitive. SAC 2003, LNCS 3006, pp. 234-247, Springer 2004.*
4. *Ross Anderson, Eli Biham and Lars Knudsen: Serpent: A Proposal for the Advanced Encryption Standard.*
5. I. Blake, X. Gao, R. Mullin, S. Vanstone and T. Yaghoobian, *Applications of Finite Fields,* Kluwer Academic Publishers, 1992.
6. Anne Canteaut, Marion Videau: *Degree of composition of highly nonlinear functions and applications to higher order differential cryptanalysis,* Eurocrypt 2002, LNCS 2332, Springer.
7. Joo Yeon Cho and Josef Pieprzyk; *Algebraic Attacks on SOBER-t32 and SOBER-128,* will appear in FSE 2004, LNCS, Springer.
8. Don Coppersmith, Shmuel Winograd: *Matrix multiplication via arithmetic progressions,* J. Symbolic Computation (1990), 9, pp. 251-280.
9. Nicolas Courtois: *Feistel Schemes and Bi-Linear Cryptanalysis,* in Crypto 2004, LNCS 3152, pp. 23-40, Springer, 2004. The extended version is available at `eprint.iacr.org/2005/251/`.
10. Nicolas Courtois: *The security of Hidden Field Equations (HFE)*; Cryptographers' Track Rsa Conference 2001, LNCS 2020, Springer, pp. 266-281.
11. Nicolas Courtois: *Algebraic Attacks on Combiners with Memory and Several Outputs,* ICISC 2004, LNCS, to appear in Springer in early 2005. Extended version available on `http://eprint.iacr.org/2003/125/`.
12. Nicolas Courtois: *La sécurité des primitives cryptographiques basées sur les problèmes algébriques multivariables MQ, IP, MinRank, et HFE*, PhD thesis, Paris 6 University, September 2001, in French. Available at `http://www.minrank.org/phd.pdf`.

13. Nicolas Courtois, Adi Shamir, Jacques Patarin, Alexander Klimov, *Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations*, In Advances in Cryptology, Eurocrypt'2000, LNCS 1807, Springer, pp. 392-407.
14. Nicolas Courtois: *Fast Algebraic Attacks on Stream Ciphers with Linear Feedback,* Crypto 2003, LNCS 2729, pp: 177-194, Springer.
15. Nicolas Courtois: *The Inverse S-box, Non-linear Polynomial Relations and Cryptanalysis of Block Ciphers,* in AES 4 Conference, Bonn May 10-12 2004, LNCS 3373, pp. 170-188, Springer, 2005.
16. Nicolas Courtois and Jacques Patarin, *About the XL Algorithm over $GF(2)$,* Cryptographers' Track RSA 2003, LNCS 2612, pages 141-157, Springer 2003.
17. 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/`.
18. Nicolas Courtois: *Higher Order Correlation Attacks, XL algorithm and Cryptanalysis of Toyocrypt,* ICISC 2002, LNCS 2587, pp. 182-199, Springer.
19. Nicolas Courtois and Willi Meier: *Algebraic Attacks on Stream Ciphers with Linear Feedback,* Eurocrypt 2003, Warsaw, Poland, LNCS 2656, pp. 345-359, Springer. An extended version is available at `http://www.minrank.org/toyolili.pdf`
20. Nicolas Courtois, Magnus Daum and Patrick Felke: *On the Security of HFE, HFEv- and Quartz,* PKC 2003, LNCS 2567, Springer, pp. 337-350. The extended version can be found at `http://eprint.iacr.org/2002/138/`.
21. Joan Daemen, Vincent Rijmen: *AES proposal: Rijndael,* `http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf`
22. Joan Daemen, Vincent Rijmen: *The Design of Rijndael. AES - The Advanced Encryption Standard,* Springer-Verlag, Berlin 2002. ISBN 3-540-42580-2.
23. Joan Daemen, Vincent Rijmen, Bart Preneel, Anton Bosselaers, Erik De Win: *The Cipher SHARK,* FSE 1996, Springer.
24. 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`
25. 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.
26. Niels Ferguson, Richard Schroeppel and Doug Whiting: *A simple algebraic representation of Rijndael,* SAC 2001, page 103, LNCS 2259, Springer.
27. Jovan Dj. Golic: *On the Security of Nonlinear Filter Generators,* FSE'96, LNCS 1039, Springer, pp. 173-188.
28. C. Harpes, G. Kramer, and J. Massey: *A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma,* Eurocrypt'95, LNCS 921, Springer, pp. 24-38. `http://www.isi.ee.ethz.ch/~harpes/GLClong.ps`
29. Philip Hawkes, Gregory Rose: *Rewriting Variables: the Complexity of Fast Algebraic Attacks on Stream Ciphers,* by Philip Hawkes and Gregory G. Rose. In crypto 2004, to appear in LNCS, Springer, 2004. Available from `eprint.iacr.org/2004/081/`.
30. Thomas Jakobsen and Lars Knudsen: *Attacks on Block Ciphers of Low Algebraic Degree,* Journal of Cryptology 14(3): 197-210 (2001).
31. Thomas Jakobsen: *Higher-Order Cryptanalysis of Block Ciphers.* Ph.D. thesis, Dept. of Math., Technical University of Denmark, 1999.
32. Thomas Jakobsen: *Cryptanalysis of Block Ciphers with Probabilistic Non-Linear Relations of Low Degree,* Crypto 98, LNCS 1462, Springer, pp. 212-222, 1998.

33. Antoine Joux, Jean-Charles Faugère: *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases*, Crypto 2003, LNCS 2729, pp. 44-60, Springer, 2003.
34. Lars R. Knudsen, Matthew J. B. Robshaw: *Non-Linear Characteristics in Linear Cryptoanalysis,* Eurocrypt'96, LNCS 1070, Springer, pp. 224-236, 1996.
35. Dong Hoon Lee, Jaeheon Kim, Jin Hong, Jae Woo Han and Dukjae Moon: *Algebraic Attacks on Summation Generators,* on `eprint.iacr.org/2003/229/` and to appear in FSE 2004, LNCS, Springer.
36. R. Lidl, H. Niederreiter: *Finite Fields*, Encyclopedia of Mathematics and its applications, Volume 20, Cambridge University Press.
37. Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone: *Handbook of Applied Cryptography*; CRC Press, 1996.
38. Tsutomu Matsumoto, Hideki Imai: *Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption,* Eurocrypt'88, Springer 1998, pp. 419-453.
39. Will Meier, Enes Pasalic and Claude Carlet: *Algebraic Attacks and Decomposition of Boolean Functions,* Eurocrypt 2004, pp. 474-491, LNCS 3027, Springer, 2004.
40. S. Murphy, M. Robshaw: *Essential Algebraic Structure within the AES,* Crypto 2002, Springer.
41. Kaisa Nyberg: *Differentially Uniform Mappings for Cryptography,* Eurocrypt'93, LNCS 765, Springer, pp. 55-64.
42. Jacques Patarin: *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*; Crypto'95, Springer, LNCS 963, pp. 248-261, 1995.
43. Jacques Patarin: *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymm. Algorithms,* Eurocrypt'96, Springer, pp. 33-48.
44. Adi Shamir, Aviad Kipnis: *Cryptanalysis of the HFE Public Key Cryptosystem*; In Advances in Cryptology, Proceedings of Crypto'99, Springer, LNCS. The paper can be found at `http://www.hfe.info`.
45. Claude Elwood Shannon: *Communication theory of secrecy systems*, Bell System Technical Journal 28 (1949), see in particular page 704.
46. Wang, D. *Elimination Methods,* Texts and Monographs in Symbolic Computation, Springer, 2001. XIII, ISBN 3-211-83241-6.