Open problems related to algebraic attacks on stream ciphers

Anne Canteaut*
INRIA - projet CODES
B.P. 105
78153 Le Chesnay cedex - France
e-mail: Anne.Canteaut@inria.fr

Abstract

The recently developed algebraic attacks apply to all keystream generators whose internal state is updated by a linear transition function, including LFSR-based generators. Here, we describe this type of attacks and we present some open problems related to its complexity. We also investigate the design criteria which may guarantee a high resistance to algebraic attacks for a keystream generator based on a linear transition function.

1 Introduction

In an additive stream cipher, the ciphertext is obtained by adding bitwise the plaintext to a pseudo-random sequence called the keystream. The keystream generator is a finite state automaton whose initial internal state is derived from the secret key and from a public initial value by a key-loading algorithm. At each time unit, the keystream digit produced by the generator is obtained by applying a *filtering function* to the current internal state. The internal state is then updated by a *transition function*. Both filtering function and transition function must be chosen carefully in order to make the underlying cipher resistant to known-plaintext attacks. In particular, the filtering function must not leak too much information on the internal state and the transition function must guarantee that the sequence formed by the successive internal states has a high period.

Stream ciphers are mainly devoted to applications which require either an exceptional encryption rate or an extremely low implementation cost in hardware. Therefore, a linear transition function seems to be a relevant choice as soon as the filtering function breaks the inherent linearity. Amongst all possible linear transition functions, those based on LFSRs are very popular because they are appropriate for

*This work was supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

low-cost hardware implementations, produce sequences with good statistical properties and can be easily analyzed. LFSR-based generators have been extensively studied. It is known that the involved filtering function must satisfy some well-defined criteria (such as a high nonlinearity, a high correlation-immunity order,...), and the designers of such generators now provide evidence that their ciphers cannot be broken by the classical attacks.

However, the recent progress in research related to algebraic attacks, introduced by Courtois and Meier [8], seems to threaten all keystream generators based on a linear transition function. In this context, it is important to determine whether such ciphers are still secure or not. Here, we investigate some related open problems, concerning the complexity of algebraic attacks (and of their variants) and concerning the design criteria of LFSR-based stream ciphers which guarantee a high resistance to these cryptanalytic techniques.

2 Basic principle of algebraic attacks

Here, we focus on binary keystream generators based on a linear transition function, which can be described as follows. We denote by $\mathbf{x}_t$ the n-bit internal state of the generator at time t. The filtering function f is first assumed to be a Boolean function of n variables, i.e., that at time t the generator outputs only one bit, $s_t = f(\mathbf{x}_t)$. The transition function is supposed to be *linear* and is denoted by $L : \mathbf{F}_2^n \to \mathbf{F}_2^n$. Therefore, we have

$$s_t = f(L^t(\mathbf{x}_0)) \ ,$$

where $\mathbf{x}_0$ is the initial state. We only consider the case where both the filtering function and the transition function are publicly known, i.e., independent from the secret key.

The basic principle of algebraic attacks goes back to Shannon's work: these techniques consist in expressing the whole cipher as a large system of multivariate algebraic equations, which can be solved to recover the secret key. A major parameter which influences the complexity of such an attack is then the degree of the underlying algebraic system. When the transition is linear, any keystream bit can obviously be expressed as a function of degree $\deg(f)$ in the initial state bits. Therefore, it is known for a long time that the filtering function involved in such a stream cipher must have high degree.

However, as pointed out by Courtois and Meier [8], the keystream generator may be vulnerable to algebraic attacks even if the degree of the algebraic function is high. Actually, the attack applies as soon as there exist relations of low degree between the output and the inputs of the filtering function f. Such relations correspond to low degree multiples of f, i.e., to relations $g(x)f(x) = h(x)$ for some g where h has a low degree. But it was proved in [13, 14] that, in the case of algebraic attacks over $\mathbf{F}_2$, the existence of any such relation is equivalent to the existence of a low degree function in the annihilator ideal of f or of $(1 + f)$. Indeed, if $g(x)f(x) = h(x)$ with $\deg(h) \leq d$, we obtain, by multiplying this equation by $f(x)$, that

$$g(x)\,[f(x)]^2 = h(x)f(x) = g(x)f(x) = h(x) \ ,$$

leading to $h(x)\,[1 + f(x)] = 0$.

Let $AN(f)$ denote the annihilator ideal of f, $AN(f) = \{g \mid g(x)f(x) = 0,\ \forall x\}$. Since the keystream bit at time t is defined by $s_t = f \circ L^t(\mathbf{x_0})$, we deduce that:

- if $s_t = 1$, any function g in $AN(f)$ leads to $g \circ L^t(\mathbf{x_0}) = 0$ for all $t \geq 0$;
- if $s_t = 0$, any function g' in $AN(1+f)$ leads to $g' \circ L^t(\mathbf{x_0}) = 0$ for all $t \geq 0$.

Therefore, if we collect the relations provided by all functions of degree at most d in $AN(f) \cup AN(f+1)$ for N known keystream bits, we obtain a system of equations of degree d depending on n variables, which correspond to the bits of the initial state. Solving such a multivariate polynomial system is a typical problem studied in Algebraic Geometry and Commutative Algebra. The main known algorithms (from the simplest to the most efficient one) are the method of linearization, XL, Buchberger algorithms, F_4 and F_5. These algorithms have different time complexities and they do not require the same number of independent equations. Even if they are essential ingredients of algebraic attacks, these algorithms are not investigated here (see e.g. [12, 4] for recent results on the relationships between these techniques).

In order to get a rough estimate of the complexity of algebraic attacks for determining the suitable parameters for the keystream generator, we only focus on the simplest technique, called *linearization*. It consists in identifying the system with a linear system of $\sum_{i=1}^{d} \binom{n}{i}$ variables, where each product of i bits of the initial state ($1 \leq i \leq d$) is seen as a new variable. The entire initial state is then recovered by a Gaussian reduction (or by more sophisticated techniques) whose time complexity is roughly

$$\left(\sum_{i=1}^{d} \binom{n}{i}\right)^{\omega} \simeq n^{\omega d}\ ,$$

where ω is the exponent of the matrix inversion algorithm, i.e., $\omega \simeq 2.37$ [6].

However, the system can be solved only if we are able to collect $\sum_{i=1}^{d} \binom{n}{i}$ linearly independent equations. The problem of determining the rank of the set $g \circ L^t(x)$ for all $0 \leq t < N$ and all g of degree at most d in $AN(f) \cup AN(1+f)$ is still open. For instance, a function g in $AN(f)$ may be invariant under composition by the transition function. Another open problem related to the complexity of the attack is to determine the proportion of monomials of degree at most d that appear in the system. This proportion may vary with the transition function.

3 Complexity of algebraic attacks

The relevant parameter in the context of algebraic attack, called the *algebraic immunity* of the filtering function, $AI(f)$, is the lowest degree achieved by a function in $AN(f) \cup AN(1+f)$. From a cryptanalytic point of view, the algebraic immunity seems more important than the number of functions with a given degree in $AN(f) \cup AN(1+f)$, which only influences the number of keystream bits required for the attack, and not the time-complexity (except maybe for some refinements such as fast algebraic attacks).

The set $AN(f)$ of all annihilating functions of f is obviously an ideal in the ring of all Boolean functions, and it is generated by $(1+f)$. It consists of the

$2^{2^n - wt(f)}$ functions of n variables which vanish on the support of f, i.e., on all x such that $f(x) = 1$, where $wt(f)$ denotes the size of the support of f. The number of functions of degree at most d in $AN(f)$ is equal to 2^κ where κ is the dimension of the kernel of the matrix obtained by restricting the Reed-Muller code of length 2^n and order d to the support of f. In other words, the rows of this matrix correspond to the evaluations of the monomials of degree at most d on $\{x, f(x) = 1\}$. Since this matrix has $\sum_{i=0}^{d} \binom{n}{i}$ rows and $wt(f)$ columns, its kernel is non-trivial when

$$\sum_{i=0}^{d} \binom{n}{i} > wt(f) \ .$$

Similarly, $AN(1 + f)$ contains some functions of degree d or less if

$$\sum_{i=0}^{d} \binom{n}{i} > 2^n - wt(f) \ .$$

Thus, as pointed out in [10], the algebraic immunity of an n-variable function is related to its Hamming weight. Most notably, for odd n, only balanced functions can have optimal algebraic immunity. A trivial corollary is also that, for any n-variable Boolean function, we have $AI(f) \leq \lceil n/2 \rceil$.

Another interesting property is that the highest possible algebraic immunity for a function is related to the number of its 0-linear structures. Let $\mathcal{S}_0(f)$ be the set of all 0-linear structures for f, i.e., $\mathcal{S}_0(f) = \{a \in \mathbf{F}_2^n, f(x + a) = f(x), \ \forall x\}$. Then,

$$AI(f) \leq \left\lceil \frac{n - \dim(\mathcal{S}_0(f))}{2} \right\rceil \ .$$

This bound is important for instance in the case of filtered LFSRs, since the filtering function usually depends only on a small subset of the internal state bits.

We deduce from the previous discussion that if an m-variable Boolean function is used for filtering the n-bit internal state of the generator, the complexity of the algebraic attack will be at most $n^{\frac{\omega m}{2}}$. This value must be higher than the complexity of an exhaustive search for the key. We here suppose that the size of the internal state is minimal with respect to key-size k, i.e. that $n = 2k$ (it is known that the size of the internal state must be at least twice the key size in order to resist time-memory trade-off attacks). Therefore, we must have $(n)^{\frac{\omega m}{2}} \geq 2^k$, i.e.,

$$m \geq 0.84 \left[\frac{k}{1 + \log_2(k)} \right] \ .$$

For instance, a filter generator with a 128-bit key and a 256-bit internal state must use a filtering function of at least 16 variables. Note that the recommended number of variables is probably higher than the previous bound because more efficient techniques can be used for solving the algebraic system.

4 Algebraic immunity of balanced functions

For 5-variable functions, it is possible to compute the algebraic immunity of all Boolean functions using the classification due to Berlekamp and Welch (because

algebraic immunity is invariant under composition by a linear permutation). We here focus on balanced functions because they are the only ones that may have optimal algebraic immunity for n odd. We can compute the algebraic immunity of all $601,080,390$ balanced functions of 5 variables:

$AI(f)$	1	2	3
nb. of balanced f	62	403,315,208	197,765,120
proportion of balanced f	10^{-7}	0.671	0.329

Similar simulations can be performed as far as the functions of n variables are classified into equivalence classes under composition by a linear permutation. But, such a classification only exist for $n = 6$ and for cubic functions up to 8 variables.

Even if some well-known constructions of cryptographic Boolean functions have been proved to have a low algebraic immunity, probabilistic arguments tend to show that the proportion of balanced functions with low algebraic immunity is very small. It has been proved in [14] that the probability that a balanced function of n variables has algebraic immunity less than $0.22n$ tends to zero when n tends to infinity. An upper bound on the probability that a balanced function has an annihilator of degree less than d is also given. This bound involves a part of the weight enumerator of $RM(d, n)$ and any new information on its complete weight distribution can clearly improve the result. However, this bound does not say anything on the average value of the algebraic immunity or on the proportion of balanced functions with optimal algebraic immunity.

The proportion of balanced functions with optimal algebraic immunity obviously corresponds to the probability that a subset of 2^{n-1} columns of the Reed-Muller code of length 2^n and of order $\lceil n/2 \rceil$ has maximal rank. If we assumed that the generator matrices of the Reed-Muller codes behave like random matrices, we would deduce that the probability that a balanced function has optimal algebraic immunity is (almost) constant. More precisely, it would be deduced for n even, that the probability that $AN(f)$ has minimal degree $n/2$ is almost 1 and, for n odd, that the probability that $AN(f)$ has minimal degree $\frac{n+1}{2}$ (resp. $\frac{n-1}{2}$) is 0.289 (resp. 0.711). One can first observe a difference with the results obtained for $n = 5$, which is not very surprising because $RM(2, 5)$ does not behave like a random code (its weight distribution is clearly not close to the distribution expected for a random code with similar parameters). Moreover, simulations tend to show that the situation differs very much from the expected one. Actually, the proportion of balanced functions of n variables with optimal algebraic immunity seems to decrease when n increases, and the average value of the algebraic immunity appears to decrease with n.

Algebraic immunity and other cryptographic criteria. Besides the Hamming weight of the function, its nonlinearity is also related to its algebraic immunity [10]. It can be proved that, for any linear function φ, the algebraic immunity of $f + \varphi$ is at most $AI(f) + 1$. Therefore, any function f of n variables with algebraic immunity at least d satisfies

$$\mathcal{NL}(f) \geq \sum_{i=0}^{d-2} \binom{n}{i} .$$

It follows that any function with optimal algebraic immunity has a high nonlinearity, more precisely

$$\mathcal{NL}(f) \geq \begin{cases} 2^{n-1} - \binom{n}{\frac{n-1}{2}} & \text{if } n \text{ is odd} \\ 2^{n-1} - \frac{1}{2}\binom{n}{\frac{n}{2}} - \binom{n}{\frac{n}{2}-1} & \text{if } n \text{ is even} \end{cases}$$

A high nonlinearity and a high algebraic immunity are then compatible criteria. Another important consequence is that the nonlinearity of a function may be a sufficient criterion to decide whether it has low algebraic immunity (but the converse is not true).

Another cryptographic property that implies that a function does not have a maximal algebraic immunity is the notion of *normality*. A function is said to be k-normal (resp. k-weakly normal) if there exists an affine subspace of dimension k on which the function is constant (resp. affine). Since the minimum weight codewords of $RM(r,n)$ are those whose support is an affine subspace of dimension $n-r$, we deduce that any k-normal function f of n variables has algebraic immunity at most $n-k$. Similarly, any k-weakly normal function has algebraic immunity at most $n-k+1$. Non-normal (and non-weakly normal) functions may be good candidates if we want to construct functions with optimal nonlinearity.

The existence of links between algebraic immunity and other cryptographic criteria remains unknown. For instance, the relation between the distance of a function to all low-degree functions (i.e., its distance to $R(d,n)$) and its algebraic immunity is still unclear. Correlation-immunity does not seem to be a priori incompatible with optimal algebraic immunity: there exists a 1-resilient function of 5 variables with optimal algebraic immunity. However, the link with all known criteria must be investigated further.

Algebraic immunity of known constructions. Some bounds have been established on the algebraic immunity of the cryptographic functions obtained by applying some classical constructions.

First, the algebraic immunity of a function can be derived from the algebraic immunities of its restrictions to a given hyperplane and to its complement [10]. For instance, if

$$f(x_1,\ldots,x_n) = (1+x_n)f_1(x_1,\ldots,x_{n-1}) + x_n f_2(x_1,\ldots,x_{n-1}) \ ,$$

we have:

- if $AI(f_1) \neq AI(f_2)$, then $AI(f) = \min(AI(f_1), AI(f_2)) + 1$;
- if $AI(f_1) = AI(f_2)$, then $AI(f) \in \{AI(f_1), AI(f_1)+1\}$.

Therefore, it is obvious how to construct a function of $2t$ variables with optimal algebraic immunity from two functions of $(2t-1)$ variables with respective algebraic immunities equal to t and to $(t-1)$. But, constructing a function of $(2t+1)$ variables with optimal algebraic immunity from two functions of $2t$ variables is much more difficult since both restrictions must have optimal algebraic immunity and they must also satisfy some additional conditions.

Some bounds on the algebraic immunities of some classical constructions, such as the Maiorana-McFarland family, can be found in [14, 10]. Moreover, an iterative construction which provides an infinite family of balanced Boolean functions with optimal algebraic immunity is presented in [11].

Computing the algebraic immunity of a Boolean function. The basic algorithm for computing the algebraic immunity of an n-variable function consists in performing a Gaussian elimination on the generator matrix of the punctured $RM(\lfloor \frac{n-1}{2} \rfloor, n)$ restricted to the support of f. This matrix has $wt(f)$ columns and $k(\lfloor \frac{n-1}{2} \rfloor, n) = \sum_{i=0}^{\lfloor \frac{n-1}{2} \rfloor} \binom{n}{i}$ rows. Therefore, the algorithm requires $k^2(\lfloor \frac{n-1}{2} \rfloor, n)wt(f)$ operations, which is close to 2^{3n-3} when f is balanced. As noted in [14], the complexity can be significantly reduced if we only want to check whether a function has annihilators of small degree d, since we do not need to consider all positions in the support of f. Indeed, considering a number of columns which is only slightly higher that the code dimension $k(d, n)$ is usually sufficient for proving that a function does not admit any annihilator of degree d.

A technique for reducing the size of the matrix over which the Gaussian elimination is performed is presented in [14]. The idea is that the elements in the support of f with low Hamming weight provide simple equations that can be removed from the matrix by a substitution step. However, due to the lack of simulation results, it is very hard to evaluate the time complexity of the substitution step in practice.

Gröbner bases algorithms such as F5 provide other techniques for computing the size of the annihilator ideal. But they need to be compared with the basic techniques in this particular context.

5 Resistance to fast algebraic attacks

At CRYPTO 2003, Courtois presented some important improvements on algebraic attacks, called *fast algebraic attacks* [7]. The refinement first relies on the existence of some low degree relations between the bits of the initial state and not only one but several consecutive keystream bits. In other word, the attacker wants to find some low degree relations g between the inputs and outputs of the function

$$\begin{array}{rccl} F_m\colon & \mathbf{F}_2^n & \to & \mathbf{F}_2^m \\ & x & \mapsto & ((f(x), f(L(x)), \ldots, f(L^m(x)) \end{array}$$

where L is the linear transition function. This function is very similar to the so-called *augmented function* defined in [1]. The fact that the augmented function may be much weaker than the filtering function, i.e, than F_0 with the previous notation, has been pointed out by Anderson [1] in the context of (fast) correlation attacks. It is an open problem to determine whether there exist relationships between the algebraic immunity of f and the algebraic immunity of F_m. Moreover, finding the low degree relations between the n inputs and m outputs of F_m becomes infeasible when m increases. The direct algorithm used for a function S with n inputs and m outputs consists in finding the low degree annihilators for the characteristic function Φ_S of S, which is the Boolean function of $(n + m)$ variables defined by

$$\Phi_S(x_1, \ldots, x_n, y_1, \ldots, y_m) = 1 \text{ if and only if } y_i = S_i(x_1, \ldots, x_n),\ \forall i\ .$$

Due to its high complexity, it can only be used for small values of m.

Since the computation of low degree relations involving several keystream bits is usually infeasible, Courtois proposed to focus on particular subclasses of relations that can be obtained much faster. The relations considered in the attack are given by linear combinations of relations of the form

$$g(x_0, \ldots, x_{\ell-1}, s_t, \ldots, s_{t+m})$$

where the terms of highest degree do not involve any keystream bits. Then, an additional precomputation step consists in determining the linear combinations of the previous relations which cancel out the highest degree monomials. Some algorithms for this step have been proposed in [7, 2]. This technique helps to decrease the degree of the relations used in the attack for different practical examples. But, here again, we do not have any theoretical result connecting the algebraic immunity of the function and the existence of such low degree linear combinations.

6 Using more sophisticated filtering functions

Many stream ciphers do not use a simple Boolean filtering function; they prefer more sophisticated mappings in order to render the attacks more difficult or in order to increase the throughput of the generator.

Multi-output Boolean functions. A basic technique for increasing the speed of the generator consists in using a filtering function with several outputs. Such functions are called vectorial Boolean functions, or *S(ubstitution)-boxes* by analogy with block ciphers. But, as pointed out in [15], the resistance of the generator to fast correlation attacks usually decreases with the number of output bits of the function. For a single output function, the attack exploits the fact that the output may be approximated by an affine function of the input variables. But, for a function S with m outputs, the attacker can apply any Boolean function g of m variables to the output vector $(y_1, \ldots, y_m)$ and he or she can perform the attack on the resulting sequence $z = g(y_1, \ldots, y_m)$. Therefore, the relevant parameter is not the nonlinearity of the vectorial function, which is the lowest Hamming distance between any linear combination of the components of S and the affine functions, but the so-called *unrestricted nonlinearity* [5], which is the lowest distance between any function $g \circ S$ and the affine functions, where g varies in the set of all nonzero Boolean functions of m variables.

For similar reasons, the algebraic immunity of a vectorial function tends to decrease with the number of output bits. For an S-box with n inputs and m outputs, there exists a relation of degree at most d in the input variables (and of any degree in the output variables) if

$$\sum_{i=0}^{d} \binom{n}{i} > 2^{n-m} .$$

A particular case of generators based on multi-output Boolean functions are the word-oriented ciphers. In order to increase the performance of software implementations, many ciphers use LFSRs over an extension field $\mathbf{F}_{2^m}$ and the associated

filtering function is usually a mapping from $\mathbf{F}_{2^m}^n$ into $\mathbf{F}_{2^m}$. This technique is used for instance in the stream cipher SNOW-v2.0, in the SOBER family and in Turing. The associated filtering function can obviously be seen as a vectorial Boolean function with mn inputs and m outputs. Consequently, all results previously mentioned apply, but the major open issue here is to determine whether word-oriented attacks can be mounted which exploit the particular structure of the function defined as a polynomial over $\mathbf{F}_{2^m}$.

Functions with memory. In some keystream generators, the filtering function is replaced by a finite automaton with some memory bits. An example is the E_0 keystream generator used in the Bluetooth wireless LAN system, which uses a combining function with 4 inputs and 4 memory bits. However, (fast) algebraic attacks [3] can still be applied on such systems. Armknecht and Krause proved that, for any filtering function of n variables with M memory bits, there always exists a relation of degree at most $\lceil \frac{n(M+1)}{2} \rceil$ between $(M+1)$ consecutive output bits and the bits of the initial state, for a given initial assignment of the memory bits. Obviously, relations of lower degree may exist. For instance, the function used in E_0 provides a relation of degree 4 involving 4 consecutive output bits, which leads to an algebraic attack of running-time around 2^{67} [3]. A similar situation occurs for multi-output functions with memory [9].

The main open issue related to the use of such sophisticated functions is to improve the efficiency of the algorithms for computing their algebraic immunity for a large number of input variables. Another related open problem is to find some general constructions which guarantee a high resistance to all these attacks.

Acknowledgements

Many thanks to Daniel Augot and Matthew Parker for their contributions to this work.

References

[1] R. J. Anderson. Searching for the optimum correlation attack. In *Fast Software Encryption - FSE'94*, volume 1008 of *Lecture Notes in Computer Science*, pages 137–143. Springer-Verlag, 1995.

[2] F. Armknecht. Improving fast algebraic attacks. In *Fast Software Encryption - FSE 2004*, volume 3017 of *Lecture Notes in Computer Science*, pages 65–82. Springer-Verlag, 2004.

[3] F. Armknecht and M. Krause. Algebraic attacks on combiners with memory. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 162–176. Springer-Verlag, 2003.

[4] G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe, and M. Sugita. Comparison between XL and Gröbner basis algorithms. In *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 338–353. Springer-Verlag, 2004.

[5] C. Carlet and E. Prouff. On a new notion of nonlinearity relevant to multi-output pseudo-random generators. In *Selected Areas in Cryptography - SAC 2003*, volume 3006 of *Lecture Notes in Computer Science*, pages 291–305. Springer-Verlag, 2004.

[6] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic programming. *Journal of Symbolic Computation*, (9):251–280, 1990.

[7] N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194. Springer-Verlag, 2003.

[8] N. Courtois and W. Meier. Algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology - EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359. Springer-Verlag, 2003.

[9] N.T. Courtois. Algebraic attacks on combiners with memory and several outputs. In *ICISC 2004*, Lecture Notes in Computer Science. Springer-Verlag, 2005. Available from `http://eprint.iacr.org/2003/125/`.

[10] D.K. Dalai, K.C. Gupta, and S. Maitra. Results on algebraic immunity for cryptographically significant Boolean functions. In *Progress in Cryptology - Indocrypt 2004*, volume 1880 of *Lecture Notes in Computer Science*, pages 92–106. Springer-Verlag, 2004.

[11] D.K. Dalai, K.C. Gupta, and S. Maitra. Cryptographically significant Boolean functions: Construction and analysis in terms of algebraic immunity. In *Fast Software Encryption - FSE 2005*, Lecture Notes in Computer Science. Springer-Verlag, 2005. To appear.

[12] C. Diem. The XL algorithm and a conjecture from commutative algebra. In *Advances in Cryptology - ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 323–337. Springer-Verlag, 2004.

[13] J.-C. Faugère and G. Ars. An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases. Technical Report 4739, INRIA, 2003. Available at `ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-4739.pdf`.

[14] W. Meier, E. Pasalic, and C. Carlet. Algebraic attacks and decomposition of Boolean functions. In *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 474–491. Springer-Verlag, 2004.

[15] M. Zhang and A. Chan. Maximum correlation analysis of nonlinear S-boxes in stream ciphers. In *Advances in Cryptology - CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 501–514. Springer-Verlag, 2000.