Fast Correlation Attacks: Methods and Countermeasures

Willi Meier

FHNW, Switzerland

Abstract. Fast correlation attacks have considerably evolved since their first appearance. They have lead to new design criteria of stream ciphers, and have found applications in other areas of communications and cryptography.

In this paper, a review of the development of fast correlation attacks and their implications on the design of stream ciphers over the past two decades is given.

Keywords: stream cipher, cryptanalysis, correlation attack.

1 Introduction

In recent years, much effort has been put into a better understanding of the design and security of stream ciphers. Stream ciphers have been designed to be efficient either in constrained hardware or to have high efficiency in software. A synchronous stream cipher generates a pseudorandom sequence, the keystream, by a finite state machine whose initial state is determined as a function of the secret key and a public variable, the initialization vector. In an additive stream cipher, the ciphertext is obtained by bitwise addition of the keystream to the plaintext.

We focus here on stream ciphers that are designed using simple devices like linear feedback shift registers (LFSRs). Such designs have been the main target of correlation attacks. LFSRs are easy to implement and run efficiently in hardware. However such devices produce predictable output, and cannot be used directly for cryptographic applications. A common method aiming at destroying the predictability of the output of such devices is to use their output as input of suitably designed non-linear functions that produce the keystream. As the attacks to be described later show, care has to be taken in the choice of these functions. Another well known method to destroy the linearity property of LFSRs is to use irregular clocking, where the output of an LFSR clocks one or more other LFSRs. All these are quite classical concepts. However they still form a valuable model for recent designs, as the hardware oriented finalists of the eSTREAM project illustrate, [41].

Several different cryptanalytic methods can be applied against stream ciphers. Amongst these methods, some only work for a specific cipher, whereas quite a number of other methods are more general, including correlation attacks, linear

A. Joux (Ed.): FSE 2011, LNCS 6733, pp. 55–67, 2011.

attacks, algebraic attacks, time/memory/data tradeoff attacks, and resynchronization attacks. We restrict here mainly to (fast) correlation attacks, and we comment on linear attacks. Beyond stream ciphers, methods similar to fast correlation attacks are of interest, e.g., in satellite communications, in the construction of a trapdoor stream cipher, [15], in digital watermarking [48], or for the learning parity with noise problem, [16], [28]. The appearance of correlation attacks has motivated various countermeasures in the form of criteria for Boolean functions that should be chosen in order to provide some correlation immunity.

This review is organized as follows. Section 2 describes the principles of correlation attacks. Section 3 forms the main part, and describes different types of fast correlation attacks. Sections 4 and 5 are aiming at countermeasures against these attacks: Section 4 discusses correlation immune functions and Bent functions, whereas Section 5 briefly deals with combiners with memory. In Section 6, linear attacks are discussed. They are viewed as a generalization of correlation attacks, and can be efficient in quite general stream cipher constructions. Finally, a few open problems are stated.

2 Correlation Attacks

The main targets of correlation attacks are filter generators and combiner generators. In a classical filter generator, the running device is a single binary LFSR. The keystream is generated as the output of a nonlinear Boolean function whose inputs are prespecified stages of a LFSR. The initial state of the LFSR is derived from the secret key and the initialization vector. In a nonlinear combiner generator, the keystream is generated as the output of a Boolean function whose inputs are the outputs of several LFSRs. In more detail, suppose the outputs $a_i^{(k)}$ of s LFSRs, $1 \leq k \leq s$, are used as input of a Boolean function f to produce keystream bits z_i for $i = 1, 2, \ldots,$

$$f(a_i^{(1)}, \ldots a_i^{(s)}) = z_i.$$

Then the keystream sequence may be correlated to the output sequence of one or more of the LFSRs.

Example 1. Let $s = 3$, and let f be the majority function,

$$y = f(x_1, x_2, x_3) = x_1x_2 + x_1x_3 + x_2x_3.$$

Then $\text{Prob}(y = x_k) = 0.75$ for $k = 1, 2, 3$.

In general, if such correlations exist, decoding techniques may be used to determine the state of the LFSRs in a divide-and-conquer manner. This is the subject of correlation attacks.

The original correlation attack was proposed by Th. Siegenthaler in [45]. Hereby, it is assumed that some portion of the keystream is known. Suppose furthermore that the keystream sequence is correlated to the output of a LFSR, i.e., $P(a_i = z_i) \neq 0.5$, where a_i and z_i are the i-th output symbols of the LFSR

and of the keystream generator, respectively. Besides the feedback connection of the LFSR, no further knowledge is required on the explicit structure of the generator.

Let the LFSR-length be n. For each of the 2^n possible initial states of the LFSR, the output sequence $\mathbf{a} = (a_1, a_2, .., a_L)$ for a suitable length $L > n$ is generated, and the value α, defined as $\alpha = L - d_H(\mathbf{a}, \mathbf{z})$ is computed. Here $d_H(\mathbf{a}, \mathbf{z})$ denotes the Hamming distance between $\mathbf{a}$ and $\mathbf{z}$, i.e., the number of positions in which $\mathbf{a}$ and $\mathbf{z}$ are different.

Then it is shown in [45], that α will take the largest value for the correct initial state with high probability, provided L in dependence of the correlation probability is sufficiently large.

This concept can be generalized to the situation where the keystream sequence is correlated to the outputs of a set of more than one LFSR: Assume that a keystream sequence is generated by a generator with several different LFSR's, and that a subset of LFSR-outputs are correlated to the keystream sequence. Then one can try to find the initial states of these LFSR's in a divide-and-conquer type of attack, and to guess the remaining LFSR-states in a separate phase.

Correlation attacks are often viewed as a decoding problem. For a LFSR of length n consider all possible output sequences of a fixed length $L > n$. This set of truncated output sequences can be viewed as a linear $[n, L]$ block code [29]. Thus the LFSR sequence $\mathbf{a} = (a_1, a_2, ..., a_L)$ is interpreted as a codeword in this code, and the keystream sequence $\mathbf{z} = (z_1, z_2, .., z_L)$ as the received channel output. The problem of the attacker can now be formulated as: Given a received word $\mathbf{z} = (z_1, z_2, ..., z_L)$, find the transmitted codeword. From coding arguments [44] it follows that the length L should be at least $L_0 = L/(1-h(1-p))$ for unique decoding, where $h(1-p)$ is the binary entropy function, and $p = P(z_i = a_i)$ is the correlation probability.

3 Fast Correlation Attacks

A *fast correlation attack* is a correlation attack that is significantly faster than exhaustive search over the initial states of the target LFSR. In [32] two algorithms for fast correlation attacks are presented. Instead of exhaustive search as originally suggested in [45], the algorithms are based on using certain parity-check equations derived from the feedback polynomial of the LFSR. The algorithms have two different phases: in the first phase, a set of suitable parity-check equations is found. In the second phase, these equations are used in a fast decoding algorithm to recover the initial state of the LFSR. These algorithms have been demonstrated to be successful for quite long LFSR's ($n = 1000$ or longer), provided the number t of feedback taps is small ($t < 10$). However the algorithms fail if the LFSR has many taps. Due to these fast correlation attacks, one usually avoids using LFSR's with few feedback taps in stream cipher design. In [50], based on earlier work in [49], the linear syndrome method from coding theory is proposed for fast correlation attacks, with similar efficiency and limitations as the algorithms in [32].

The two algorithms in [32] are described here in order. In a preparation phase, parity check equations are determined by observing that for a given position j the digit a_j of the LFSR-sequence $\mathbf{a}$ satisfies a certain number m of linear relations involving a fixed number t of other digits of $\mathbf{a}$. Here t denotes the number of taps of the LFSR. These linear relations are found by shifting and iterated squaring of the LFSR-relation.

Example 2. Consider the LFSR of length $n = 3$ with feedback relation

$$a_j = a_{j-1} + a_{j-3},\ j \geq 3.$$

Then by squaring, the relation $a_j = a_{j-2} + a_{j-6}$ does hold as well. And by shifting, one gets three relations for the same digit a_j:

$$\begin{array}{rcl} a_{j-3} + a_{j-1} + a_j & = & 0 \\ a_{j-2} + a_j + a_{j+1} & = & 0 \\ a_j + a_{j+2} + a_{j+3} & = & 0 \end{array}$$

The digits of the known output sequence $\mathbf{z}$ are substituted in the linear relations thus obtained. Some of the relations will still hold, some others will not. It has been observed that the more relations are satisfied for a digit z_j, the higher is the (conditional) probability that $z_j = a_j$. Denote by p^* the probability for $z_j = a_j$, conditioned on the number of relations satisfied.

Consider first a digit contained in one relation. Assume the digit $a^{(0)} = a_j$ at a given position j satisfies a linear relation involving t digits at some other positions of the LFSR-sequence $\mathbf{a}$,

$$a^{(0)} + a^{(1)} + a^{(2)} + \cdots a^{(t)} = 0.$$

Denote by $z^{(0)}, z^{(1)}, \ldots z^{(t)}$ the digits in the same positions of the output sequence. Then

$$\begin{array}{l} z^{(0)} = a^{(0)} + b^{(0)} \\ z^{(1)} = a^{(1)} + b^{(1)} \\ \ldots\ldots\ldots\ldots\ldots \\ z^{(t)} = a^{(t)} + b^{(t)}, \end{array}$$

and for the perturbations, $\mathrm{Prob}(b^{(0)} = 0) = \ldots = \mathrm{Prob}(b^{(t)} = 0) = p$. Denote $s = \mathrm{Prob}(b^{(1)} + \ldots + b^{(t)} = 0) : s = s(p,t)$. Then $s(p,t)$ can be computed recursively:

$$s(p,1) = p,\ s(p,t) = ps \cdot (p,t-1) + (1-p)(1-s(p,t-1)) \text{ for } t > 1.$$

Next assume that a specified digit $a = a_j$ is contained in m relations, each involving t other digits. For a subset S of relations, denote by $E(S)$ the event that exactly the relations in S are satisfied. Then for $z = z_j$,

$$\begin{array}{l} \mathrm{Prob}((z = a), \text{and } E(S)) = p \cdot s^h (1-s)^{m-h}, \\ \mathrm{Prob}((z \neq a) \text{ and } E(S)) \ = (1-p) s^{m-h} (1-s)^h, \end{array}$$

where $h = |S|$ denotes the number of relations in S. Hence the conditional probability $p^* = \text{Prob}(z = a|E(S))$ is given by

$$p^* = \frac{p \cdot s^h (1-s)^{m-h}}{p \cdot s^h (1-s)^{m-h} + (1-p) s^{m-h} (1-s)^h}.$$

The probability distributions for the number h of satisfied relations are Binomial distributions. There are two cases. If the digit z is correct, i.e., if $z = a$,

$$p_1 = \binom{m}{h} s^h (1-s)^{m-h}.$$

Alternatively, if $z \neq a$,

$$p_0 = \binom{m}{h} s^{m-h} (1-s)^h.$$

It is intuitively clear that a digit a can be more reliably predicted the more the two distributions are separated. In [32] it is shown that the average number m of relations involving a that can be checked in the given output stream $\mathbf{z}$ is:

$$m = \log_2 \left(\frac{L}{2n} \right) (t+1).$$

Example 3. Assume a correlation probability $p = 0.75$, a number $t = 2$ of taps, LFSR-length $n = 100$, and a length of $L = 5000$ known bits of $\mathbf{z}$. Then $m = 12$ relations are available (in average), and $s = 0.75^2 + 0.25^2 = 0.625$. The value of the probability p^* conditioned on the number h of relations satisfied is:

relations satisfied	probability
12	0.9993
11	0.9980
10	0.9944

Based on these considerations, two algorithms, Algorithms A and B for fast correlation attacks are described in [32].

Algorithm A essentially chooses a set I_0 of approximately n digits of the known output stream $\mathbf{z}$ that satisfy the most relations. The digits in I_0 are taken as a reference guess of $\mathbf{a}$ at the same positions. Thereafter, the initial state of the LFSR is found by solving a system of linear equations.

As the selected digits in I_0 are only correct with some probability, the correct guess of the initial state is found by testing modifications of I_0 of Hamming distance $1, 2, \ldots$ by correlation of the corresponding LFSR-sequence with the given sequence $\mathbf{z}$. Thus Algorithm A has exponential complexity, of order $O(2^{cn})$, $0 < c < 1$. The parameter c is a function of the correlation p, the number of taps t, and the ratio L/n.

Example 4. Let $p = 0.75$, $t = 2$, and $L/n = 100$. Then $c = 0.012$. The search complexity is of significantly reduced order $O(2^{0.012n})$ compared to $O(2^n)$ in case of exhaustive search.

Algorithm B is described step by step as follows:

Algorithm B

1. Assign the correlation probability p to every digit of $\mathbf{z}$.
2. To every digit of $\mathbf{z}$ assign the new probability p^*. Iterate this step a number of times.
3. Complement those digits of $\mathbf{z}$ with $p^* < p_{thr}$ (for a suitable threshold p_{thr}).
4. Stop, if $\mathbf{z}$ satisfies the basic relation of the LFSR, else go to 1.

The number of iterations in 2. and the probability threshold in 3. have to be adequately chosen to obtain maximum correction effect. In 2. the formula for recomputing conditional probabilities has to be generalized to the case where assigned probabilities for each involved digit are different. After a few iterations, a strong separation effect can be observed between digits having probability p^* close to 0 or close to 1. Algorithms B is essentially linear in the LFSR-length n. The success of this algorithm has extensively been verified experimentally for various correlation probabilities, LFSR-lengths and numbers of taps $t < 10$. Iterative methods similar to Algorithm B have been applied in decoding. In [17], R. G. Gallager has developed a decoding scheme, where the decoder computes all the parity checks and then changes any digit that is contained in more than some fixed number of unsatisfied parity-check equations. Using these new values, the parity-checks are recomputed, and the process is repeated. The method in [32] contrasts to this approach in that the process of assigning conditional probabilities to every digit is iterated rather than just changing digits according to the number of parity-check equations satisfied.

As these algorithms work only if the LFSR has few feedback taps, i.e., if the feedback polynomial is of low weight, the problem persisted, how to design algorithms that are efficient even if the number of taps is arbitrary.

A first approach is to look for polynomial multiples of the feedback polynomial: If the recursion is not of low weight, consider multiples of the feedback polynomial that are of low weight.

Example 5. ([30]) Consider the connection polynomial $g(x)$ over $GF(2)$ of degree 7 and of weight 5:

$$g(x) = x^7 + x^6 + x^4 + x + 1.$$

$g(x)$ has a polynomial multiple (a trinomial)

$$f(x) = g(x)m(x) = x^{21} + x^3 + 1$$

with a polynomial $m(x)$ of degree 14.

Fast correlation attacks can likewise be applied to the linear recursion of sparse polynomial multiples, [4]. There are quite different methods on how to find low

weight polynomial multiples. These methods differ in the weight and degree of an attempted sparse multiple, and in the required memory and computing time, see, e.g., [18], [47]. In particular, a feedback polynomial of a LFSR of length n can be shown to have a polynomial multiple of weight 4 (i.e., with 3 taps) of expected length about $2^{n/3}$.

Low weight multiples of feedback polynomials are of more general interest, as they often allow for distinguishing attacks on LFSR-based stream ciphers, e.g., [23], [12], [13]. In these attacks, the primary aim is not to recover the key, but to distinguish the known keystream from random.

Apart from investigation of sparse multiples of the connection polynomial, there is vast literature dealing with improvements of the initial algorithms. A major improvement concerns fast correlation attacks on LFSR's with an arbitrary number of feedback taps. It appears that the algorithms as proposed in [24] and [35] are amongst the most efficient known thus far. Based on these methods, in [6], the algorithmic steps have been improved to accelerate the attacks in [24] or [35]. As to be expected, the complexity of these algorithms depends on the length n of the target LFSR as well as on the correlation probability p. A version of one of these algorithms is briefly sketched:

As opposed to other fast correlation attacks, the use of parity-checks is combined with a partial exhaustive search over a subset B of the initial state of the targeted LFSR. As predictions are true only with some probability, $D > n$ targeted bits of the LFSR-output are predicted by evaluating and counting a number of parity-check equations. As before, the parity-checks are found in a preprocessing phase. In [6], an elaborate match-and-sort algorithm is described how to generate many parity-checks. In an example case, the parity-checks involve a number of bits in the set B, the target bit a_i at position i of the LFSR-sequence $\mathbf{a}$ to be predicted, and two other bits at some positions j and m in the known output stream $\mathbf{z}$. The procedure is informally as follows:

- For each of the D target bits, evaluate a large number of parity-checks substituted into the output stream $\mathbf{z}$ and the guessed bits of B, and count the number of parity-checks satisfied, N_s, and the number of parity-checks N_u not satisfied.
- If the expression $|N_s - N_u|$ is larger than a threshold, predict $a_i = z_i$ if $N_s > N_u$, else $a_i = z_i + 1$.

Provided this majority poll is decisive for D target bits of the LFSR-sequence, the initial state can be easily recovered. Estimates of the complexity of this algorithm suggest that it is possible to attack LFSRs of length n about 100 in practice, provided p is not too close to 0.5. In [36] and [26], a large part of the state of the art in fast correlation attacks is found.

Fast correlation attacks have been applied successfully to concrete constructions: In [21], a fast correlation attack is applied to the summation generator. In [25], the stream cipher LILI-128 has been cryptanalysed by such methods.

More recently, in [2] the initial version of the eSTREAM finalist Grain with a key of 80 bits was broken. This motivated a careful tweak, Grain v1, which is an eSTREAM finalist, [41].

4 Towards Correlation Immunity

In many (fast) correlation attacks, the correlations are deduced as linear approximations of nonlinear output functions in stream ciphers. The existence of correlation attacks has thus led to new design criteria for Boolean functions used in stream ciphers, [46], [33]. In particular, combining (or filter) functions should have no statistical dependence between any small subset of inputs and the output.

More formally, let $X_1, X_2, \ldots X_n$ be independent binary variables which are balanced (i.e., each variable takes the values 0 and 1 with probability $\frac{1}{2}$). A Boolean function f of n variables is *m-th order correlation immune* if for each subset of m random variables $X_{i1}, X_{i2}, \ldots, X_{im}$ the random variable

$$Z = f(X_1, X_2, \ldots, X_n)$$

is statistically independent of the random vector $(X_{i1}, X_{i2}, \ldots, X_{im})$. There exists a tradeoff between the order of correlation immunity and the algebraic degree of Boolean functions, [46]. Low algebraic degree conflicts with security: Due to the Berlekamp-Massey algorithm, [31], and due to algebraic attacks, [8], the degree of output functions of combining or filter generators should not be low. Finally, to prevent good statistical approximations of the output function f by linear functions, f should have large distance to all affine functions. In this regard, early work by D. Chaum and J.-H. Evertse, [5], [14] on the cryptanalysis of the DES block cipher motivated a different trail concerning cryptographic properties of Boolean functions and S-boxes: In [33], a class of Boolean functions, coined perfect nonlinear functions, was studied, which turned out to coincide with the class of Bent functions, [42]. These functions have been used in the context of coding theory, [29]. Bent functions are not balanced, but otherwise they share a number of desirable properties: They have maximum nonlinearity, i.e., they have largest possible distance to affine functions, and they satisfy good correlation resistance. In addition, they have optimum differential properties. In a series of papers, K. Nyberg has studied Boolean functions and S-boxes related to Bent functions, starting with [39], [40]. A prominent example of such a vectorial Boolean function is the multiplicative inverse map in the finite field $GF(2^8)$ which is used in the S-box of the AES block cipher. The study of Boolean functions with good cryptographic properties has been an ongoing topic, see, e.g., the book [9].

5 Combiners with Memory

The tradeoff between correlation immunity and algebraic degree as noticed in [46] can be avoided if the combining function is allowed to have memory. Results on combiners with memory have first been published by R. Rueppel, [43].

A (k, m)-combiner with k inputs and m memory bits is a finite state machine which is defined by an output function

$$f : \{0,1\}^m \times \{0,1\}^k \to \{0,1\}$$

and a memory function

$$\varphi : \{0,1\}^m \times \{0,1\}^k \to \{0,1\}^m.$$

For a given stream $(X_1, X_2, \ldots)$ of inputs, $X_t \in \{0,1\}^k$, and an initial assignment $C_1 \in \{0,1\}^m$ of the memory, an output bitstream $(z_1, z_2, \ldots)$ is defined according to

$$z_t = f(C_t, X_t).$$

and

$$C_{t+1} = \varphi(C_t, X_t)$$

for all $t > 0$. For keystream generation, the stream of inputs $(X_1, X_2, \ldots)$ is produced by the output of k driving devices. The initial states are determined by the secret key. Often, the driving devices are LFSRs.

Example 6. The basic summation generator with $k = 2$ inputs is a combiner with $m = 1$ bit memory, which coincides with the usual carry of addition of integers: Write $X_t = (a_t, b_t)$. The functions f and φ are defined by

$$z_t = f(c_t, a_t, b_t) = a_t \oplus b_t \oplus c_t$$

and

$$c_{t+1} = \varphi(c_t, a_t, b_t) = a_t b_t \oplus a_t c_t \oplus b_t c_t.$$

The function f in this summation generator is 2^{nd}-order correlation immune. Correlations in combiners with one bit memory have been studied in detail in [34].

Example 7. The stream cipher E_0 used in Bluetooth, [3], is a combiner with $k = 4$ inputs and $m = 4$ bit memory. The stream of inputs is produced by the outputs of 4 LFSRs of length 128 in total.

More recent (word-oriented) stream ciphers with memory are, e.g., SNOW, [11], the eSTREAM finalist SOSEMANUK, [41], or ZUC, [37]. A concept related to combiners with memory are feedback with carry shift registers (FCSRs) as introduced in [27]. A synthesis based on FCSRs enabled to cryptanalyze summation generators.

6 Linear Attacks

A correlation attack may be successful, if there are found linear relations that hold with nonnegligible probabilities, between single output bits and a subset of state bits of the LFSR's involved. A *linear attack* is more general, as it seeks

for "good linear approximations" of the output stream, i.e., for correlations between linear functions of *several* output bits and linear functions of a subset of the LFSR-state bits involved. This type of attacks may be successful for both, key recovery as well as for distinguishing the output from random. Linear attacks have been developed by Golić, [19]. If there are strong enough correlations, a number of equations, each of which does hold with some probability, may be derived. There are fairly efficient methods (reminiscent to fast correlation attacks) to solve such systems of equations, provided the known output stream is long enough, i.e., provided there are many more equations than unknowns (see [22] for an attack of this type on the Bluetooth stream cipher algorithm). The distinction between correlations of a *single* output bit to linear functions of state bits of the LFSR's as opposed to correlations of linear functions of *several output bits* to linear functions of state bits of the LFSR's becomes relevant if the non-linear combining system contains m bit memory: Consider a block of M consecutive output bits, $Z_t^M = (z_t, z_{t-1}, ..., z_{t-M+1})$ as a function of the corresponding block of M consecutive inputs $X_t^M = (X_t, X_{t-1}, ..., X_{t-M+1})$ and the preceeding memory bits C_{t-M+1}. Here X_t denotes the bit vector at time t of the state bits of the LFSRs involved, and similarly, C_{t-M+1} denotes the bit vector of the m memory bits at time $t - M + 1$. Assume that X_t^M and C_{t-M+1} are balanced and mutually independent. Then, according to [20], if $M \geq m$, there *must* exist linear correlations between the output and input bits, but they may also exist if $M < m$. This shows that correlations cannot be easily defeated, even in presence of memory. Besides key recovery attacks, powerful distinguishing attacks using linear approximations of quite diverse stream cipher constructions have become known, e.g. a linear distinguisher on the initial version of the stream cipher SNOW, [7], or a distinguisher on the cipher Shannon, [1].

7 Open Problems

The topic of (fast) correlation attacks has considerably evolved over time. However, some open problems in this area are identified. A first question is how to devise fast correlation attacks in an algorithmically optimal way. Important steps in this direction have been taken in [6] and [10]. In a second direction, various word-oriented stream ciphers use LFSRs over a binary extension field of $GF(2)$ rather than over $GF(2)$. In this case, the established methods seem infeasible. It would be of interest to see fast correlation attacks for LFSRs, e.g, over $GF(2^{32})$. This question has been addressed initially in [26]. Finally, it was observed that correlations cannot be easily avoided in whatever construction is used in the design of a stream cipher. In a complexity-theoretic context, it has been shown that there exist pseudorandom generators with low computational requirements so that in a specified sense each linear distinguisher of the output stream has a bias that can provably be upper bounded, [38]. It would be interesting to come up with cryptographically secure constructions with similar properties.

Acknowledgments

This review owes much to useful discussions with María Naya-Plasencia and with attendees of FSE 2011. This work is supported by DTU Mathematics and by the Danish Center for Applied Mathematics and Mechanics as well as by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center of the Swiss National Science Foundation under grant number 5005-67322.

References

1. Ahmadian, Z., Mohajeri, J., Salmasizadeh, M., Hakala, R., Nyberg, K.: A practical distinguisher for the Shannon cipher. Journal of Systems and Software 83(4), 543–547 (2010)
2. Berbain, C., Gilbert, H., Maximov, A.: Cryptanalysis of grain. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 15–29. Springer, Heidelberg (2006)
3. Bluetooth SIG, Specification of the Bluetooth system, Version 1.1 (February 22, 2001), `http://www.bluetooth.com/`
4. Canteaut, A., Trabbia, M.: Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 573–588. Springer, Heidelberg (2000)
5. Chaum, D., Evertse, J.-H.: Cryptanalysis of DES with a Reduced Number of Rounds, sequences of linear factors in block ciphers. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 192–211. Springer, Heidelberg (1986)
6. Chose, P., Joux, A., Mitton, M.: Fast Correlation Attacks: An Algorithmic Point of View. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 209–221. Springer, Heidelberg (2002)
7. Coppersmith, D., Halevi, S., Jutla, C.S.: Cryptanalysis of stream ciphers with linear masking. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 515–532. Springer, Heidelberg (2002), `http://eprint.iacr.org/2002/020`
8. Courtois, N., Meier, W.: Algebraic attacks on Stream Ciphers with Linear Feedback. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 345–359. Springer, Heidelberg (2003)
9. Cusick, T., Stanica, P.: Cryptographic Boolean Functions and Applications. Academic Press, London (2009)
10. Edel, Y., Klein, A.: Computational aspects of fast correlation attacks (2010) (preprint)
11. Ekdahl, P., Johansson, T.: A New Version of the Stream Cipher SNOW. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 47–61. Springer, Heidelberg (2003)
12. Ekdahl, P., Meier, W., Johansson, T.: Predicting the Shrinking Generator with Fixed Connections. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 330–344. Springer, Heidelberg (2003)
13. Englund, H., Johansson, T.: A New Simple Technique to Attack Filter Generators and Related Ciphers. In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 39–53. Springer, Heidelberg (2004)
14. Evertse, J.-H.: Linear structures in block ciphers. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 249–266. Springer, Heidelberg (1988)

15. Finiasz, M., Vaudenay, S.: When Stream Cipher Analysis Meets Public-Key Cryptography. In: Biham, E., Youssef, A.M. (eds.) SAC 2006. LNCS, vol. 4356, pp. 266–284. Springer, Heidelberg (2007)
16. Fossorier, M.P.C., Mihaljević, M.J., Imai, H., Cui, Y., Matsuura, K.: An Algorithm for Solving the LPN Problem and Its Application to Security Evaluation of the HB Protocols for RFID Authentication. In: Barua, R., Lange, T. (eds.) INDOCRYPT 2006. LNCS, vol. 4329, pp. 48–62. Springer, Heidelberg (2006)
17. Gallager, R.G.: Low-Density Parity-Check Codes. MIT Press, Cambridge (1963)
18. Golić, J.: Computation of low-weight parity-check polynomials. Electronic Letters 32(21), 1981–1982 (1996)
19. Golić, J.: Linear models for keystream generators. IEEE Trans. on Computers 45, 41–49 (1996)
20. Golić, J.: Correlation properties of a general binary combiner with memory. Journal of Cryptology 9, 111–126 (1996)
21. Golić, J., Salmasizadeh, M., Dawson, E.: Fast correlation attacks on the summation generator. Journal of Cryptology 13, 245–262 (2000)
22. Golić, J.D., Bagini, V., Morgari, G.: Linear Cryptanalysis of Bluetooth Stream Cipher. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 238–255. Springer, Heidelberg (2002)
23. Hawkes, P., Rose, G.: Exploiting Multiples of the Connection Polynomial in Word-Oriented Stream Ciphers. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 303–316. Springer, Heidelberg (2000)
24. Johansson, T., Jönsson, F.: Fast correlation attacks through reconstruction of linear polynomials. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 300–315. Springer, Heidelberg (2000)
25. Jönsson, F., Johansson, T.: A fast correlation attack on LILI-128. Inf. Process. Lett. 81(3), 127–132 (2002)
26. Jönsson, F.: Some results on fast correlation attacks, Thesis, Lund University, Sweden
27. Klapper, A., Goresky, M.: Feedback shift registers, 2-adic span, and combiners with memory. Journal of Cryptology 10, 111–147 (1997)
28. Levieil, É., Fouque, P.-A.: An Improved LPN Algorithm. In: De Prisco, R., Yung, M. (eds.) SCN 2006. LNCS, vol. 4116, pp. 348–359. Springer, Heidelberg (2006)
29. MacWilliams, F.J., Sloane, N.J.: The theory of error-correcting codes. North Holland, Amsterdam (1977)
30. Maitra, S., Gupta, K.C., Venkateswarlu, A.: Results on multiples of primitive polynomials and their products over GF(2). Theor. Comput. Sci. 341(1-3), 311–343 (2005)
31. Massey, J.L.: Shift-register synthesis and BCH decoding. IEEE Transactions on Information Theory 15, 122–127 (1969)
32. Meier, W., Staffelbach, O.: Fast correlation attacks on certain stream ciphers. Journal of Cryptology 1, 159–176 (1989)
33. Meier, W., Staffelbach, O.: Nonlinearity criteria for cryptographic functions. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 549–562. Springer, Heidelberg (1990)
34. Meier, W., Staffelbach, O.: Correlation properties of combiners with memory in stream ciphers. Journal of Cryptology 5, 67–86 (1992)
35. Mihaljević, M.J., Fossorier, M.P.C., Imai, H.: Fast correlation attack algorithm with list decoding and an application. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 196–210. Springer, Heidelberg (2002)

36. Mihaljevic, M., Fossorier, M., Imai, H.: On decoding techniques for cryptanalysis of certain encryption algorithms. IEICE Trans. Fundamentals E84-A(4), 919–930 (2001)
37. Mobile Phone Security Algorithms - New Version, http://gsmworld.com/our-work/programmes-and-initiatives/
38. Mossel, E., Shpilka, A., Trevisan, L.: On ϵ-biased Generators in NC^0. Random Struct. Algorithms 29(1), 56–81 (2006)
39. Nyberg, K.: Constructions of Bent Functions and Difference Sets. In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 151–160. Springer, Heidelberg (1991)
40. Nyberg, K.: Perfect Nonlinear S-Boxes. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 378–386. Springer, Heidelberg (1991)
41. Robshaw, M., Billet, O.: New Stream Cipher Designs. LNCS, vol. 4986. Springer, Heidelberg (2008)
42. Rothaus, O.S.: On bent functions. Journal of Combinatorial Theory (A) 20, 300–305 (1976)
43. Rueppel, R.A.: Correlation immunity and the summation generator. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 260–272. Springer, Heidelberg (1986)
44. Shannon, C.E.: Communication theory of secrecy systems. Bell System Technical Journal 27, 656–715 (1949)
45. Siegenthaler, T.: Decrypting a class of stream ciphers using ciphertext only. IEEE Trans. on Computers C-34, 81–85 (1985)
46. Siegenthaler, T.: Correlation immunity of nonlinear combining functions for cryptographic applications. IEEE Trans. Inform. Theory 30, 776–780 (1984)
47. Wagner, D.: A generalized birthday problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–303. Springer, Heidelberg (2002)
48. Wang, D., Lu, P.: Geometrically Invariant Watermark Using Fast Correlation Attacks. In: Proceedings of IIH-MSP 2006, pp. 465–468. IEEE Computer Society, Los Alamitos (2006)
49. Zeng, K., Huang, M.: On the linear syndrome method in cryptoanalysis. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 469–478. Springer, Heidelberg (1990)
50. Zeng, K., Yang, C.H., Rao, T.R.N.: An improved linear syndrome algorithm in cryptanalysis with applications. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 34–47. Springer, Heidelberg (1991)