Fast correlation attacks against stream ciphers and related open problems

Anne Canteaut
INRIA - projet Codes
B.P. 105
78153 Le Chesnay cedex - France
Email: Anne.Canteaut@inria.fr

***Abstract*— Fast correlation attacks have been considerably improved recently, based on efficient decoding algorithms dedicated to very large linear codes in the case of a highly noisy channel. However, a better adaptation of these techniques to the concrete involved stream ciphers is still an open issue.**

I. Introduction

In an additive synchronous stream cipher, the ciphertext is obtained by adding bitwise the plaintext to a pseudo-random sequence called the keystream. This keystream is generated by a finite state automaton whose initial state is derived from the secret key, and usually 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 secure. In particular, the filtering function must not leak too much information on the internal state and the transition function must guarantee that, for (almost) all initial states, 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 in software or an extremely low implementation cost in hardware (see e.g. [1]). These implementation constraints influence the design choices, especially for the transition function. Keystream generators can be divided into the following main families depending on the procedure used for updating the internal state:

- *generators based on a linear transition function.* A linear transition function seems to be a relevant choice for hardware implementation as soon as the filtering function breaks the inherent linearity. Amongst all possible linear transition functions, those based on linear feedback shift registers (LFSRs) are very popular because they are appropriate for low-cost hardware implementations, produce sequences with good statistical properties and can be easily analyzed.
- *generators based on a nonlinear transition function.* The weaknesses resulting from the linearity of the transition function, especially the vulnerability to algebraic attacks [2], can be avoided by choosing a nonlinear transition mapping. However, for hardware-oriented stream ciphers, this function must guarantee that the generated sequence has a high period. This condition may be avoided when the size of the internal state is not limited by implementation constraints: in that case, the probability that a short cycle exists is very low because of the large size of the internal state (e.g. RC4 [3]). But, for hardware applications, the internal state cannot be much larger than the bound provided by time-memory-data tradeoff attacks, i.e., twice the key size. Therefore, theoretical results on the period of the sequence generated by the transition function are required. Only a few appropriate mappings can be used in this context, such as Feedback with Carry Shift Registers [4], Nonlinear Feedback Shift Registers, T-functions [5]...
- *Hybrid transition functions.* In some keystream generators, the internal state is split into two parts: the first one is updated linearly and the other one has a nonlinear behavior. When the nonlinear part is much smaller than the linear one, it is usually identified with internal memory; for instance, SNOW 2.0 [6] or E0 [7] are viewed as LFSR-based stream ciphers with memory. But, some keystream generators such as PANAMA [8] or MUGI [9] use linear and nonlinear parts of similar sizes.

From the cryptanalyst's point of view, the trend towards splitting the internal state of the keystream generator into different parts suggests the use of divide-and-conquer attacks. The *correlation attack*, which was originally proposed by Siegenthaler against combination generators [10], applies when a part of the internal state is updated independently from the other ones and has a reasonable size. This attack has been greatly improved by Meier and Staffelbach [11], [12] when the target part of the internal state is updated linearly. In this case, efficient error-correcting decoding can be used in order to (partially) recover the initial state of the generator.

II. Correlation attack

Here, we focus on binary keystream generators 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 assumed to be a Boolean function of n variables: at time t the generator outputs a single bit, $s_t = f(\mathbf{x}_t)$. In order to produce an unbiased sequence, f must obviously be balanced, i.e., it must output 0 or 1 with probability $1/2$. The transition

function is denoted by $\Phi : \mathbf{F}_2^n \to \mathbf{F}_2^n$. Therefore, we have

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

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.

We investigate known-plaintext attacks which aim at recovering the initial state $\mathbf{x}_0$, which is in that sense identified with the key of the cipher. However, it must be pointed out that the initial state is usually computed from a shorter secret key and from a public initial value. Some additional information on $\mathbf{x}_0$ can therefore be derived, especially in the context of related IVs attacks.

A correlation attack, as originally described by Siegenthaler against combination generators, can actually be mounted as soon as the n-bit internal state $\mathbf{x}_t$ of the generator can be decomposed into two parts $\mathbf{y}_t$ and $\mathbf{z}_t$ of respective sizes ℓ and $n-\ell$, which are updated independently from each other, i.e.,

$$(\mathbf{y}_{t+1}, \mathbf{z}_{t+1}) = (\Phi_1(\mathbf{y}_t), \Phi_2(\mathbf{z}_t)) \ .$$

The attack aims at recovering one of the parts of the initial state, e.g. $\mathbf{y}_0$, called the target state. The attack applies if and only if there exists a Boolean function g of ℓ variables which is correlated to the filtering function f. This equivalently means that there exists g from $\mathbf{F}_2^\ell$ into $\mathbf{F}_2$ such that

$$p_g = P_{Y,Z}[f(Y,Z) = g(Y)] > \frac{1}{2}$$

where Y and Z are two independent random variables uniformly distributed in $\mathbf{F}_2^\ell$ and $\mathbf{F}_2^{n-\ell}$.

If such a function g exists, the target sequence $\boldsymbol{\sigma} = (\sigma_t)_{t\geq 0}$ defined by $\sigma_t = g(\Phi_1^t(\mathbf{y}_0))$ is correlated to the keystream sequence $\mathbf{s} = (s_t)_{t\geq 0}$. This correlation can be detected by computing the correlation between N bits of the keystream and the corresponding bits of the target sequence $\boldsymbol{\sigma}(\mathbf{y}_0)$ generated from the initial state $\mathbf{y}_0$:

$$C(\mathbf{s}, \boldsymbol{\sigma}(\mathbf{y}_0)) = \sum_{t=0}^{N-1} (-1)^{s_t \oplus \sigma_t(\mathbf{y}_0)} \ .$$

The expected value of this quantity is equal to $2N(p_g - \frac{1}{2})$ when $\mathbf{y}_0$ is the correct value of the target initial state. Therefore, the attack consists of an exhaustive search for the target ℓ-bit part of the initial state $\mathbf{y_0}$. For each possible value of $\mathbf{y_0}$, the first N bits of the corresponding sequence $\boldsymbol{\sigma}(\mathbf{y_0})$ are computed and the correlation with the known keystream $\mathbf{s}$ is evaluated. A right guess for $\mathbf{y_0}$ can be distinguished from a wrong one by comparing $C(\mathbf{s}, \boldsymbol{\sigma}(\mathbf{y_0}))$ to a given threshold. For a wrong guess, both sequences $\mathbf{s}$ and $\boldsymbol{\sigma}(\mathbf{y_0})$ are actually expected to be uncorrelated. This procedure can be seen as a basic statistical test for distinguishing two binary random sources: one distributed according to the uniform distribution, the other one according to the distribution of $f(Y,Z) \oplus g(Y)$, i.e., $P[X=1] = p_g$ [13], [14]. When p_g is close to $\frac{1}{2}$, the attack requires the knowledge of

$$N = \mathcal{O}\left(\left(p_g - \frac{1}{2}\right)^{-2}\right)$$

keystream bits. The time complexity for recovering the ℓ-bit target part of the initial state is therefore

$$\mathcal{O}\left(2^\ell \left(p_g - \frac{1}{2}\right)^{-2}\right) \ .$$

From the cryptanalyst's point of view, this attack raises the question of the optimal choice for the function g. For the designer, the underlying problem consists in finding the filtering functions f which make the attack infeasible. Both questions can be answered by computing the probability that an ℓ-variable function g coincides with f. This probability involves the distributions of the output of f when its first ℓ inputs are fixed, namely $p_y = P_Z[f(y,Z)=1]$. Actually, we have

$$\begin{aligned} p_g &= \frac{1}{2^\ell}\left(\sum_{y\in g^{-1}(0)} (1-p_y) + \sum_{y\in g^{-1}(1)} p_y\right) \\ &= \frac{1}{2} + \frac{1}{2^\ell}\sum_{y\in\mathbf{F}_2^\ell} (-1)^{g(y)}\left(\frac{1}{2} - p_y\right) \end{aligned} \quad (1)$$

In the attack, g must then be chosen such that p_g is maximal; this situation occurs if and only if all terms in the above sum are positive, i.e., if

$$\begin{cases} g(y) = 1 & \text{if } p_y > \frac{1}{2} \\ g(y) = 0 & \text{if } p_y < \frac{1}{2} \end{cases}$$

It follows that

$$\max_g p_g = \frac{1}{2} + \frac{1}{2^\ell}\sum_{y\in\mathbf{F}_2^\ell} \left|\frac{1}{2} - p_y\right| \ .$$

Therefore, it clearly appears that the correlation attack can be prevented if the filtering function f is such that its output remains uniformly distributed when its first ℓ input variables are fixed. Such functions are said to be *resilient* (or *correlation-immune*) with respect to its first ℓ variables. More generally, the correlation-immunity order of a function, defined by Siegenthaler [15], is the highest number of variables ℓ such that the output distribution of the function is unchanged when any ℓ inputs are fixed. In the special case of a combination generator where the inputs of f correspond to the outputs of m independent LFSRs, the minimum number of LFSRs which must be considered together in a correlation attack is $\ell+1$ where ℓ is the correlation-immunity order of f.

III. Fast correlation attacks as a decoding problem

One major problem in correlation attacks is that they perform an exhaustive search for an entire part of the initial state, leading to a huge time-complexity. The fast correlation attacks introduced by Meier and Staffelbach [12] considerably reduce the running-time but require a longer segment of known keystream. They apply when the target sequence $\boldsymbol{\sigma}$ generated by

$$\sigma_t = g(\Phi_1^t(\mathbf{y}_0))$$

depends linearly on the ℓ-bit target initial state $\mathbf{y}_0$. In this case, any N-bit subsequence of $\boldsymbol{\sigma}$ can be seen as a codeword of a linear code $\mathcal{C}$ of length N and dimension ℓ. The attack aims at recovering the codeword corresponding to $\boldsymbol{\sigma}$ from the knowledge of N consecutive keystream bits where

$$p = P[\sigma_t \neq s_t] < \frac{1}{2}, \text{ for all } t \ .$$

The key idea of fast correlation attacks consists in viewing the correlation attack as a decoding problem: the keystream subsequence $(s_t)_{t<N}$ can be seen as the result of the transmission of $(\sigma_t)_{t<N}$ through a binary symmetric channel (BSC) with cross-over probability p. Thus, recovering the target part of the initial state, $\mathbf{y}_0$, consists in decoding the keystream subsequence relatively to the linear code $\mathcal{C}$.

This formulation enables to derive a lower bound on the keystream length required for a successful attack. Actually, a linear code of dimension ℓ can be successfully decoded only if its rate does not exceed the capacity of the transmission channel [16]. Here, the capacity of the binary symmetric channel with cross-over probability p is given by $C(p) = 1 + p\log_2 p + (1-p)\log_2(1-p)$. As in most practical situations, the error-probability p is very close to $1/2$, namely $p = 1/2 - \varepsilon$, the capacity can be approximated by $C(1/2-\varepsilon) \simeq 2\varepsilon^2/\ln(2)$, leading to the following required keystream length:

$$N \geq \frac{\ell}{C(p)} \simeq \frac{\ln(2)\ell}{2\varepsilon^2} \ .$$

Therefore, fast correlation attacks are based on fast decoding procedures for the linear code $\mathcal{C}$ of length N and dimension ℓ, when N must be as close as possible to Shannon's limit.

IV. Correlation attacks on some basic LFSR-based generators

(Fast) correlation attacks have been originally described against LFSR-based generators, which is still a major family of synchronous stream ciphers. In this case, the target part of the initial state corresponds to one (or a few) LFSR involved in the system.

A. Fast correlation attacks on combination generators

In the case of a combination generator, it can be proved that the best target sequence $\boldsymbol{\sigma}$ is the sequence obtained by adding the outputs of $(\ell+1)$ constituent LFSRs, where ℓ is the correlation-immunity order of the combining function [17], [18]. This result directly comes from the fact that the $(\ell+1)$-variable function g which maximizes (1) is the sum of all its inputs (up to the addition of a binary constant). Thus, the target sequence $\boldsymbol{\sigma}$ corresponds to the output of a unique LFSR whose feedback polynomial is the greatest common divisor of the feedback polynomials of the $(\ell+1)$ involved LFSRs. Since the feedback polynomials are usually chosen to be primitive, the length of the target LFSR is the sum of the lengths of the $(\ell+1)$ LFSRs. The keystream corresponds to the received word as output of the binary symmetric channel with cross-over probability

$$p = Pr[s_t \neq \sigma_t] = \frac{1}{2} - \frac{1}{2^{m+1}} \max_{u \in \mathbf{F}_2^m, wt(u)=\ell+1} |\hat{\mathrm{f}}(u)| \ ,$$

where m is the number of variables of the combining function (i.e., the number of constituent LFSRs), ℓ is its correlation-immunity order and $\hat{\mathrm{f}}$ denotes the Walsh transform of f.

B. Fast correlation attacks on filter generators

Since fast correlation attacks avoid an exhaustive search for the target part of the initial state, they may also be successfully applied to the entire initial state. They can then be mounted on filter generators whose internal state consists of a single LFSR. Here, the keystream corresponds to the output of f when its inputs are some fixed taps of the LFSR, i.e.,

$$s_t = f(v_{t+\gamma_1}, v_{t+\gamma_2}, \ldots, v_{t+\gamma_m}), \ \forall t \geq 0 \ ,$$

where $\mathbf{v}$ is the LFSR-sequence. Thus, the target sequence $\boldsymbol{\sigma}$ is produced by an LFSR which has the same feedback polynomial as the constituent LFSR, but a different initial state. The optimal target sequence $\boldsymbol{\sigma}$ actually corresponds to

$$\sigma_t = \sum_{i=1}^{m} \alpha_i v_{t+\gamma_i}$$

where $\alpha = (\alpha_1, \ldots, \alpha_m)$ is the vector which maximizes the magnitude of the Walsh transform of the filtering function. Thus, the cross-over probability can be estimated by

$$p = P[s_t \neq \sigma_t] = \frac{\mathcal{NL}(f)}{2^m} \ ,$$

where $\mathcal{NL}(f)$ is the nonlinearity of the filtering function and m is the number of its input variables.

However, the binary symmetric channel model which was originally described for combination generator does not hold anymore. The reason is that the inputs of the filtering function at different times are not independent, even if these dependences are usually reduced by an appropriate choice of the input taps $(\gamma_i)_{1\leq i\leq m}$ [19]. In the case of filter generators, the underlying transmission channel is not memoryless. Actually, simulations show that the performance of fast correlation attacks against practical filter generators is usually worse than expected from the BSC model [20]. But, finding the appropriate channel model and dedicated efficient decoding procedures is still an open problem.

The dependences between the successive inputs of the filtering function can be exploited by applying more sophisticated correlation attacks. A first improvement consists in considering correlations involving several consecutive keystream bits. Instead of studying the properties of f, the attack relies on the so-called augmented function defined by

$$\begin{array}{rccc} F_k\colon & \mathbf{F}_2^{\ell} & \rightarrow & \mathbf{F}_2^{k} \\ & y & \mapsto & (f(y), f(\Phi(y)), \ldots, f(\Phi_1^{k-1}(y))) \end{array}$$

As pointed out by Anderson [21], the augmented function may present much larger correlations than the original filtering

function. However, the complexity for finding the optimal correlations between the ℓ inputs and k outputs of F_k highly increases with k. It is an open problem to find either an efficient algorithm or new theoretical results for computing these correlations. A related issue is the existence of some relationships between the correlation properties of the augmented function and those of f. Another technique for increasing the involved correlation consists in conditioning the correlation by the observed value of some keystream bits. Combining both methods leads to block-oriented conditional correlation attacks as described in [22]–[24].

A completely different improvement of fast correlation attacks on filter generators consists in exploiting several linear approximations of the keystream together [25], even all linear functions which are correlated to f [26]. Then, we get a larger number of linear relations, leading to a more efficient decoding. The main modification is that the involved transmission channel is now a non-stationary channel, since the different linear approximations of f do not correspond to the same error-probability. However, it can be proved that the capacity of this channel is much larger than the capacity of the channel corresponding to the best linear approximation. Moreover, it only depends on the number of nonzero Walsh coefficients of the filtering function [26] and it is not related to its nonlinearity.

C. LFSR-based generators with memory

Most recent proposals of LFSR-based generators include some memory bits, i.e., a small part of the internal state which is not linearly updated. Similar attacks can be mounted for instance against combiners with memory, as shown in [27]. A nice example of such attacks is the recent cryptanalysis of E0 presented by Lu and Vaudenay [28], [29].

In most previous cases, the target sequence $\boldsymbol{\sigma}$ is a linear recurring sequence; it can be generated by a LFSR of length ℓ whose feedback polynomial is known to the attacker. In this situation, the linear code $\mathcal{C}$ associated to $\boldsymbol{\sigma}$ has a very particular algebraic structure. A generator matrix for this code is the $\ell \times N$-matrix $G = (g_{i,j})$ whose t-th column is given by

$$\sum_{i=0}^{\ell-1} g_{i,t} X^i = X^t \bmod P^\star(X) \ ,$$

where $P^\star$ is the characteristic polynomial of the involved LFSR, i.e., the reciprocal polynomial of its feedback polynomial. This structure is extensively exploited by some decoding procedures, especially by iterative decoding algorithms.

V. Decoding techniques for fast correlation attacks

A. Maximum-likelihood decoding

When it is seen as a decoding problem, the original correlation attack proposed by Siegenthaler consists in applying a maximum-likelihood decoding algorithm to the code $\mathcal{C}$ associated to $\boldsymbol{\sigma}$. The number N of required keystream bits is equal to Shannon's bound.

The basic algorithm for maximum-likelihood decoding consists in computing the Hamming distance between the N-bit received word (i.e., the keystream subsequence) and all codewords. Its complexity is then $N2^\ell$. But, when $\mathcal{C}$ is a linear code, this algorithm is equivalent to the computation of the Fourier transform of the ternary function F from $\mathbf{F}_2^\ell$ into $\{-1, 0, 1\}$ defined by

$$\left\{ \begin{array}{rcll} F(g_t) & = & (-1)^{s_t} & \text{for all } 0 \le t < N \\ F(x) & = & 0 & \text{for all } x \notin \{g_t,\ 0 \le t < N\} \end{array} \right.$$

where g_t is the t-th column of the generator matrix of $\mathcal{C}$. Actually, the correlation between the keystream and the target sequence $\boldsymbol{\sigma}(\mathbf{y_0})$ generated from a given initial state $\mathbf{y_0}$ corresponds to the Fourier coefficient of F at point $\mathbf{y_0}$:

$$\begin{aligned} C(\mathbf{s}, \boldsymbol{\sigma}(\mathbf{y}_0)) &= \sum_{t=0}^{N-1} (-1)^{s_t \oplus \mathbf{y}_0 \cdot g_t} \\ &= \sum_{x \in \mathbf{F}_2^\ell} F(x)(-1)^{\mathbf{y}_0 \cdot x} = \widehat{F}(\mathbf{y}_0) \ . \end{aligned}$$

For large values of N, the time complexity of the algorithm can then be reduced to $\ell 2^\ell$ by using a fast Fourier transform algorithm [30]. However, decoding becomes infeasible for practical LFSR lengths, namely $\ell \ge 80$. For this reason, the attacker needs to use much faster decoding algorithms. But, no efficient general decoding algorithm is known for achieving the channel capacity. This means that practical fast correlation attacks require that the known running-key sequence be much longer than Shannon's lower bound.

The fast decoding algorithms used in correlation attacks can be divided into two families. The first one consists of decoding procedures which make use of the inherent structure of the code, especially when it corresponds to a LFSR. The second family contains general algorithms that can be applied to any linear code.

B. Decoding based on low-weight parity-check equations

Most decoding algorithms which exploit the structure of the generator matrix use the existence of sparse parity-check equations for the linear code $\mathcal{C}$. This technique was first proposed by Meier and Staffelbach in their original paper [12] and later improved [17]. It especially applies when the target sequence $\boldsymbol{\sigma}$ corresponds to the output of a LFSR with feedback polynomial P. Actually, any sparse multiple $1 + X^{a_1} + \ldots + X^{a_{d-1}}$ of the feedback polynomial P exactly corresponds to a linear relation involving d bits of the LFSR sequence:

$$\forall t, \ \ \sigma_t \oplus \sigma_{t-a_1} \oplus \cdots \oplus \sigma_{t-a_{d-1}} = 0 \ .$$

When a collection of such sparse equations is available, the LFSR code can be viewed as a low-density parity-check (LDPC) code. Then, some efficient iterative decoding techniques [31] can be applied to those codes. The starting point of these procedures is the information on σ_t derived from the keystream, namely

$$Obs(\sigma_t) = Pr[\sigma_t = 1 | \mathbf{s}] = \left\{ \begin{array}{ll} p & \text{if } s_t = 0 \\ 1-p & \text{if } s_t = 1 \end{array} \right.$$

Then, the algorithm computes the extrinsic information on each σ_t in its m-th parity-check equation, $\sigma_t = \bigoplus_{j \in J_m} \sigma_j$:

$$Ext_m(\sigma_t) = P[\bigoplus_{j \in J_m} \sigma_j = 1 | \mathbf{s}] \ .$$

We can then derive an a posteriori probability (APP) on σ_t by

$$APP(\sigma_t) \propto Obs(\sigma_t) \prod_{m=1}^{m_d} Ext_m(\sigma_t) \ .$$

Iterative algorithms consist in updating the extrinsic information $Ext_m(\sigma_t)$ by using the partial APPs (i.e., APPs excluding the extrinsic information given by the m-th equation). In the context of fast correlation attacks, the original belief propagation algorithm cannot be applied because of its high complexity, both in time and memory. In practice, the extrinsic information is updated from the values of the total APPs (instead of the partial APPs). Some additional approximations for computing these values are usually used, leading to faster algorithms, even if they may be less efficient in terms of error-correcting capability.

When parity-check equations with d terms are used, the required keystream length is given by

$$N \propto \left(\frac{1}{2\varepsilon}\right)^{\frac{2(d-2)}{d-1}} 2^{\frac{\ell}{d-1}} \ ,$$

where $p = \frac{1}{2} - \varepsilon$ is the cross-over probability of the channel. The precomputation step (i.e., the search for the parity-check equations) requires $\mathcal{O}\left(N^{d-2}/(d-2)!\right)$ operations, but this complexity can be reduced by using a time-memory trade-off technique or the algorithm proposed in [30]. However, this complexity implies that the weight of the equations cannot increase very much. Finally, the complexity of the decoding step is roughly

$$\left(\frac{1}{2\varepsilon}\right)^{\frac{2d(d-2)}{d-1}} 2^{\frac{\ell}{d-1}} \ .$$

Other decoding algorithms based on low-density parity-check equations are described in [32]–[34].

C. General decoding algorithms

The other class of algorithms can be applied to any linear code. The key-idea due to Chepyshov, Johansson and Smeets [35] is the following: when the code dimension ℓ is too large for ML-decoding, it is possible to derive from the original code a new code with smaller dimension on which ML-decoding can be applied. Obviously, a code of dimension $k < \ell$ is obtained from the columns of the generator matrix which vanish of the last $\ell - k$ positions. However, since the number of such columns (i.e., the new code length) is very small, ML-decoding requires a huge keystream length. In order to increase the code length, we then include in the generator matrix of the new code every linear combination of w columns of the original matrix which vanishes of the last $(\ell - k)$ positions. Now, the new code has dimension k and length

$$\frac{N^w}{w!\ 2^{\ell-k}} \ .$$

For reasonable values of k, it can be decoded with ML-decoding if

$$N \geq \left(\frac{1}{2\varepsilon}\right)^2 2^{\frac{\ell-k}{w}} \ .$$

The precomputation step (construction of the new generator matrix), when linear combinations of w columns are used, is very similar to the computation of parity-check equations of weight $w+1$ for LDPC decoding. But, the decoding complexity is now very different. Here, the first k bits of the target initial state $\mathbf{y}_0$ are recovered with $k2^k$ operations (with a FFT). The other ones can be recovered by successively applying the same technique but for the next k bits of $\mathbf{y}_0$.

Some improvements of the previous algorithm are presented in [36]. The first one considers all linear combinations of w columns of the generator matrix whose $(\ell - k)$ last positions lie in a given subset (the original algorithm proposed in [35] corresponds to the case where this subset is reduced to the zero vector). This refined version is very similar to the extended version of linear cryptanalysis on block ciphers based on multiple approximations [37]. A second improvement is derived from an algorithm due to Goldreich, Rubinfeld and Sudan [38] for polynomial reconstruction in the multivariate case. Instead of repeating the procedure for several independent k-bit blocks of the ℓ-bit target initial state, it uses a sequential procedure for recovering the other $(\ell - k)$ bits one after the other.

In practice, the most efficient fast correlation attacks enable to recover a target initial state of length 60 for an error-probability $p = 0.4$ in a few hours on a PC from the knowledge of 10^6 keystream bits [39]. However, it is worth noticing that the number of known keystream bits required by all these decoding methods is still exponential in the size of the target part of the initial state, even if the gain compared to Siegenthaler's attack is very high in practice.

VI. Adapting the decoding algorithms to practical situations

The main promising research directions for improving fast correlation attacks consist in adapting the usual decoding algorithms to the practical context of a given stream cipher. Even if the underlying problem corresponds to a classical decoding problem, the involved parameters do not fit in with the common situation. Fast correlation attacks could then be significantly improved by using some decoding methods which suits better to the involved cipher than to the BSC model.

A first approach would be to make a better use of soft-decoding algorithms. Besides the observation of the keystream, some additional soft input information may be derived from the structure of the cipher, especially from the key-loading and the IV-loading procedures. It is worth noticing that the entropy of the initial state does not exceed the size of the key, which is at least twice smaller than the internal state. As previously noted, considering several keystream bits together by means of the augmented function may also provide such a soft information on the target sequence. A soft output may also be of interest when the decoding step does not succeed,

since it could be given as input to another type of attack. The sequential use of several decoding techniques may also be considered in this context.

Another important open issue is the decoding of linear codes over a large binary alphabet. In order to increase the performance of software implementations, most recent keystream generators 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 $\mathbf{F}_{2^m}$. The typical alphabet size is then 2^{32} for word-oriented stream ciphers (e.g. [6]). The attacks corresponding to direct adaptations of the usual decoding algorithms become then infeasible for such a huge alphabet. Moreover, finding the appropriate target sequence is still a very hard problem in this case. Attacking such word-oriented ciphers then requires much more efficient algorithms both for decoding and for computing the correlations induced by the filtering function.

VII. Acknowledgments

The author would like to thank Marion Videau for helpful discussions. This work is partially supported by the European Commission via ECRYPT network of excellence IST-2002-507932.

References

[1] European Network of Excellence in Cryptology - ECRYPT, "ECRYPT stream cipher project," http://www.ecrypt.eu.org/stream/, 2005.

[2] N. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback," in *Advances in Cryptology - EUROCRYPT 2003*, ser. LNCS, vol. 2656, 2003, pp. 345–359.

[3] R. Rivest, "The RC4 encryption algorithm," RSA Data Security, 1992.

[4] F. Arnault and T. Berger, "F-FCSR: Design of a new class of stream ciphers," in *Fast Software Encryption - FSE 2005*, ser. LNCS, vol. 3557. Springer-Verlag, 2005, pp. 83–97.

[5] A. Klimov and A. Shamir, "A new class of invertible mappings," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, ser. LNCS, vol. 2523. Springer-Verlag, 2003, pp. 470–483.

[6] P. Ekdahl and T. Johansson, "A new version of the stream cipher SNOW," in *Selected Areas in Cryptography - SAC 2002*, ser. LNCS, vol. 2295. Springer-Verlag, 2002, pp. 47–61.

[7] "Bluetooth specifications – version 1.0b," http://www.bluetooth.com.

[8] J. Daemen and C. Clapp, "Fast hashing and stream encryption with PANAMA," in *Fast Software Encryption - FSE'98*, ser. LNCS, vol. 1372. Springer-Verlag, 1998, pp. 60–74.

[9] Hitachi, Ld, "MUGI specification," http://www.sdl.hitachi.co.jp/crypto/mugi/mugi_spe.pdf, 2001.

[10] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Computers*, vol. C-34, no. 1, pp. 81–84, 1985.

[11] W. Meier and O. Staffelbach, "Fast correlation attacks on stream ciphers," in *Advances in Cryptology - EUROCRYPT'88*, ser. LNCS, vol. 330. Springer-Verlag, 1988, pp. 301–314.

[12] ——, "Fast correlation attack on certain stream ciphers," *J. Cryptology*, pp. 159–176, 1989.

[13] T. Cover and J. Thomas, *Elements of Information theory*. Wiley, 1991.

[14] P. Junod, "Statistical cryptanalysis of block ciphers," Ph.D. dissertation, École Polytechnique Fédérale de Lausanne, Switzerland, 2005.

[15] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications," *IEEE Trans. Inform. Theory*, vol. 30, no. 5, pp. 776–780, 1984.

[16] C. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, 1948.

[17] A. Canteaut and M. Trabbia, "Improved fast correlation attacks using parity-check equations of weight 4 and 5," in *Advances in Cryptology - EUROCRYPT 2000*, ser. LNCS, vol. 1807. Springer-Verlag, 2000, pp. 573–588.

[18] M. Zhang, "Maximum correlation analysis of nonlinear combining functions in stream ciphers," *J. Cryptology*, vol. 13, no. 3, pp. 301–313, 2000.

[19] J. Golić, "On the security of nonlinear filter generators," in *Fast Software Encryption - FSE'96*, ser. LNCS, vol. 1039. Springer-Verlag, 1996, pp. 173–188.

[20] S. Leveiller, "Quelques algorithmes de cryptanalyse du registre filtré," Ph.D. dissertation, École Nationale Supérieure des Télécommunications, France, 2004, in French.

[21] R. J. Anderson, "Searching for the optimum correlation attack," in *Fast Software Encryption - FSE'94*, ser. LNCS, vol. 1008. Springer-Verlag, 1995, pp. 137–143.

[22] S. Lee, S. Chee, S. Park, and S. Park, "Conditional correlation attack on nonlinear filter generators," in *Advances in Cryptology - ASIACRYPT'96*, ser. LNCS, vol. 1163. Springer-Verlag, 1996, pp. 360–367.

[23] B. Löhlein, "Attacks based on conditional correlations against the nonlinear filter generator," IACR Preprint 2003/020, 2003, available at http://eprint.iacr.org/2003/020/.

[24] S. Leveiller, G. Zémor, P. Guillot, and J. Boutros, "A new cryptanalytic attack for PN-generators filtered by a Boolean function," in *Selected Areas in Cryptography - SAC 2002*, ser. LNCS, vol. 2596. Springer-Verlag, 2003, pp. 232–249.

[25] F. Jönsson and T. Johansson, "A fast correlation attack on LILI-128," *Information Processing Letters*, vol. 81, no. 3, pp. 127–132, Feb. 2002.

[26] A. Canteaut and E. Filiol, "On the influence of the filtering function on the performance of fast correlation attacks on filter generators," in *Proceedings of 23rd Symposium on information theory in the Benelux*, May 2002.

[27] J. Golić, "Correlation properties of a general binary combiner with memory," *J. Cryptology*, vol. 9, no. 2, pp. 111–126, 1996.

[28] Y. Lu and S. Vaudenay, "Faster correlation attack on Bluetooth keystream generator E0," in *Advances in Cryptology - CRYPTO 2004*, ser. LNCS, vol. 3152. Springer-Verlag, 2004, pp. 407–425.

[29] ——, "Cryptanalysis of Bluetooth keystream generator two-level E0," in *Advances in Cryptology - ASIACRYPT 2004*, ser. LNCS, vol. 3329. Springer-Verlag, 2004, pp. 483–499.

[30] P. Chose, A. Joux, and M. Mitton, "Fast correlation attacks: An algorithmic point of view," in *Advances in Cryptology - EUROCRYPT 2002*, ser. LNCS, vol. 2332. Springer-Verlag, 2002, pp. 209–221.

[31] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, 1962.

[32] T. Johansson and F. Jönsson, "Improved fast correlation attack on stream ciphers via convolutional codes," in *Advances in Cryptology - EUROCRYPT'99*, ser. LNCS, vol. 1592. Springer-Verlag, 1999, pp. 347–362.

[33] ——, "Fast correlation attacks based on turbo code techniques," in *Advances in Cryptology - CRYPTO'99*, ser. LNCS, vol. 1666. Springer-Verlag, 1999, pp. 181–197.

[34] M. Mihaljevic, M. Fossorier, and H. Imai, "A low-complexity and high performance algorithm for the fast correlation attack," in *Fast Software Encryption - FSE 2000*, ser. LNCS, vol. 1978. Springer-Verlag, 2000.

[35] V. Chepyshov, T. Johansson, and B. Smeets, "A simple algorithm for fast correlation attacks on stream ciphers," in *Fast Software Encryption - FSE 2000*, ser. LNCS, vol. 1978. Springer-Verlag, 2000, pp. 181–195.

[36] T. Johansson and F. Jönsson, "Fast correlation attacks through reconstruction of linear polynomials," in *Advances in Cryptology - CRYPTO'00*, ser. LNCS, vol. 1880. Springer-Verlag, 2000, pp. 300–315.

[37] A. Biryukov, C. De Cannière, and M. Quisquater, "On multiple linear approximations," in *Advances in Cryptology - CRYPTO 2004*, ser. LNCS, vol. 3152. Springer-Verlag, 2004, pp. 1–22.

[38] O. Goldreich, R. Rubinfeld, and M. Sudan, "Learning polynomials with queries: the highly noisy case," in *36th Annual Symposium on Fondation of Computer Science*, Milwaukee, Wisconsin, 1995, pp. 294–303.

[39] F. Jönsson, "Some results on fast correlation attacks," Ph.D. dissertation, Lund University, Sweden, 2002.