Decimation Attack of Stream Ciphers

Eric Filiol
Ecoles militaires de Saint-Cyr Coëtquidan,
DGER/CRECSC/DSI, 56381 Guer Cedex, FRANCE
efiliol@mailhost.esm-stcyr.terre.defense.gouv.fr

September 22, 2000

Abstract

This paper presents a new attack called *Decimation Attack* of most stream ciphers. It exploits the property that multiple clocking (or equivalently d-th decimation) of a LFSR can simulate the behavior of many other LFSRs of possible shorter length. It yields then significant improvements of all the previous known correlation and fast correlation attacks. A new criterion on the length of the polynomial is then defined to resist to the decimation attack. Simulation results and complexity comparison are detailed for ciphertext only attacks.

Keywords: stream cipher, linear feedback shift register, correlation attack, fast correlation attack, sequence decimation, multiple clocking.

1 Introduction

Despite growing importance of block ciphers, stream ciphers remain a very important class of cipher systems mainly used by governmental world.

In a binary additive stream cipher, the ciphertext is obtained by bitwise addition of the plaintext to a pseudo-random sequence called the *running key*. This latter is produced by a pseudo-random generator whose initial state constitutes the secret key. Most real-life designs center around *Linear Feedback Shift-Register* (LFSR) combined by a nonlinear Boolean function. Different variant exist: clock-controlled systems, filter generators, multiplexed systems, memory combiners, decimated generators,... This paper will focus on the most common class of combination generators depicted in Figure 1.

The cryptanalyst's problem often deals with that of recovering the initial states of some LFSRs, assuming that the structure of the generator is known to him.State of the art in generic stream ciphers cryptanalysis can be summarized

also INRIA Projet Codes Domaine de Voluceau BP 105 78153 Le Chesnay Cedex France *Eric.Filiol@inria.fr*

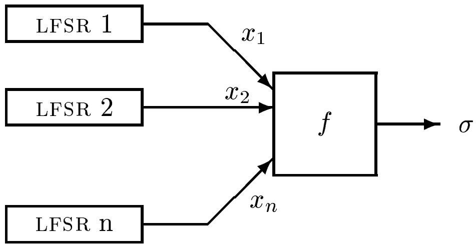

Figure 1: Nonlinear combination generator

as follows: correlation and fast correlation attacks. Both exploit an existing correlation (of order k) between the running key σ and some linear combination of the k input variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. A so-called *divide and conquer attack* is conducted which consists to try to recover the initial state of the k target LFSRs independently of the other unknown key bits. Such correlations always exist but functions offering a good cryptographic resistance generally offer a high correlation order k thus imposing on the cryptanalyst to consider simultaneously several LFSRs [6]. In this case it is obvious that the k distinct LFSRs can be seen as a unique LFSR of length L; the length and the feedback polynomial P of this LFSR can be derived from the feedback polynomials of the constituent polynomials [17]. In the following we will generalize by speaking of the unique target LFSR of length L. A single LFSR will then be a particular case for $k = 1$.

- In correlation attacks [18, 19], the $2^L - 1$ possible initializations of the LFSR are exhaustively tried and each time its corresponding produced sequence x is compared to the captured running key σ. The initialization yielding the closest awaited statistical bias is supposed to be the correct one. This attack is limited by the length L of the target LFSR (by now $L \approx 50$) since it requires $2^L - 1$ trials and practical attacks can deal with only correlation order $k = 1$ (i.e. considering only one LFSR). Moreover the longer the LFSR is and the lower the correlation value, the longer the necessary running key sequence will be. This attack is nowadays no longer efficient except for weak schemes.

- Fast correlation attacks were introduced by Meier and Staffelbach [13] and avoid examining all possible initializations of the LFSR. The output of the target LFSR is considered to have passed through a noisy channel, most frequently modelled by the Binary (Memoryless) Symmetric Channel, BSC with some error probability $p < \frac{1}{2}$, where $\epsilon = \frac{1}{2} - p$ is usually very small. In this setting, an LFSR output sequence of fixed length N can be considered as a (N, L) binary linear code. Each of its codewords can be uniquely related to a possible initialization. Then the cryptanalyst's problem becomes a decoding problem in the presence of a BSC with strong noise. Meier and Staffelbach attack uses iterative decoding process for low-density parity-check codes when feedback polynomial is of low-weight. Minor improvements have then been obtained [3, 14, 15]. Johansson and Jönsson recently presented new powerful ideas by considering convolutional codes [8] and turbo-codes [9]. Canteaut and Trabbia in [2] show that Gallager iterative decoding [7] with parity-check equations of weight 4 and 5 is usually more efficient than all other previous attacks since it successfully decodes very high error probabilities with a relatively feasible time and memory complexity.

All these attacks have the length L of the target LFSR as major limitation (particularly when the correlation order k is high). As soon as L increases too much and ϵ is too small, the memory and complexity requirements explode, making these attacks unfeasible. This paper presents how to bypass this limitation. By considering d-fold clocking (or d-th decimation) of the target LFSR output sequence, we show how to use a simulated shorter LFSR thus improving the known attacks. This approach is particularly efficient when dealing with long LFSRs or with combining functions of high correlation order of real-life designs. With a relatively longer sequence we significantly reduce the complexity and suppress the memory requirements of the best known attack (Canteaut and Trabbia [2]). Moreover this attack consider only the LFSR length and not the feedback polynomial. We will only consider ciphertext only attack to make this attack more realistic.

This paper is organized as follows. Section 2 presents the theoretical tools we use in this attack. In Section 3, the Decimation Attack (DA) itself is described and simulation results for different cases are given. A new criterion is then defined allowing to choose LFSRs resisting this attack. Section 4 compares decimation attack with the best known attack of [2].

These results can be easily generalized to most other kind of stream ciphers. Numerous details can be found at[5].

2 Theoretical background

2.1 Linear Feedback Shift Register sequences

A linear feedback shift register of length L is characterized by L binary connection coefficients $(p_i)_{1\leq i\leq L}$. It associates to any L-bit initialization $(s_t)_{1\leq t\leq L}$ a sequence $(s_t)_{t>0}$ defined by the L-th order linear recurrence relation

$$s_{t+L} = \sum_{i=1}^{L} p_i s_{t+L-i}, \quad t \geq 0 \ .$$

The connection coefficients are usually represented by a univariate polynomial P over $\mathbf{F}_2$, called *the feedback polynomial*:

$$P(X) = 1 + \sum_{i=1}^{L} p_i X^i \ .$$

Most applications use a primitive feedback polynomial to ensure that the periods of all sequences produced by the LFSR are maximal.

Let us consider the sequence σ produced by LFSR i_1, LFSR $i_2, \dots$, LFSR i_k combined by a function g. The sequence σ obvioulsy corresponds to the output of a unique LFSR of length L. The length and the feedback polynomial P of this LFSR can be determined from the feedback polynomials of the k LFSRs. We have $L \leq g(L_{i_1}, L_{i_2}, \dots, L_{i_k})$ where g is evaluated over the integers. Equality holds when the feedback polynomials of the k LFSRs are primitive and when their degrees are coprimes [17]. Generally linear approximation of the original combining function f is considered, that is to say $g = u.x$ is linear. Then any feedback polynomial of $\sigma = g(x_{i_1}, x_{i_2}, \dots, x_{i_k})$ is $\prod_{i \in supp(u)} P_i$ where $supp(u) = \{i, u_i = 1\}$.

2.2 Decimation of LFSR sequences

Let us consider a sequence $\sigma = \sigma_1, \sigma_2, \dots$, produced by a LFSR of length L whose feedback polynomial is irreducible in $GF(q)[X]$. Suppose now that we operate a sampling on σ at intervals of d clock cycles (d-fold clocking) thus producing a subsequence $\rho = \rho_1, \rho_2, \dots$. In other words, it is equivalent to the d-decimation of the original sequence σ. Thus we have $\rho_i = \sigma_{dj}$ for $j = 0, 1, 2, \dots$ or in sequence notation $\rho = \sigma[d]$.

With d-fold clocking, the original LFSR will behave like a different LFSR which is called the *simulated LFSR* [16]. The interesting question is to determine its properties, especially relatively to the original LFSR. They are summarized in the following proposition.

Proposition 1 *[16, page 146] Let σ be the sequence produced by the original LFSR whose feedback polynomial $P(x)$ is irreducible in $GF(q)$ of degree L. Let α be a root of $P(x)$ and let T be the period of $P(x)$. Let ρ the sequence resulting from the d-th decimation of σ, i.e. $\rho = \sigma[d]$. Then the simulated LFSR, that is the LFSR directly producing ρ has the following properties:*

1. *The feedback polynomial $P^*(x)$ of the simulated LFSR is the minimum polynomial of α^d in $GF(q^L)$.*
2. *The period T^* of $P^*(x)$ is equal to $\frac{T}{\gcd(d,T)}$.*
3. *The degree L^* of $P^*(x)$ is equal to the multiplicative order of q in $\mathbf{Z}_{T^*}$.*

Moreover all d in C_k where $C_k = \{k, kq, kq^2, \dots,\}$ mod T denotes the cyclotomic coset of k modulo T, result in the same simulated LFSR, except for different initial contents. Finally, every sequence producible by the simulated LFSR is equal to $\sigma[d]$ for some choice of the initial contents of the original LFSR.

In real case applications, $P(x)$ is primitive so $T = 2^L - 1$. Thus when T is not prime, by a careful choice of d such that $\gcd(2^L - 1, d) \neq 1$ one may expect to obtain a simulated LFSR shorter than the original one.

Example 1 *Let us consider the original LFSR with $P(x) = X^4 + X + 1$.*

- *$d \in C_3 = \{3, 6, 9, 12\}$, $T^* = 5$ and $L^* = 4$ since the multiplicative order of 2 in $\mathbf{Z}_5$ is 4.*

- *$d \in C_5 = \{5, 10\}$, $T^* = 3$ and $L^* = 2$ since the multiplicative order of 2 in $\mathbf{Z}_3$ is 2 (and $P^*(x) = X^2 + X + 1$).*

The feedback polynomial $P^*(x)$ of the simulated LFSR can be obtained by applying the Berlekamp-Massey LFSR synthesis algorithm [11] to the sequence ρ.

2.3 Boolean functions for stream ciphers

A Boolean function with n variables is a function from the set of n-bit words, $\mathbf{F}_2^n$, into $\mathbf{F}_2$.

The *Walsh-Hadamard transform* of a Boolean function f refers to the Fourier transform of the corresponding sign function, $x \mapsto (-1)^{f(x)}$:

$$\forall u \in \mathbf{F}_2^n, \;\; \widehat{\chi}_f(u) = \sum_{x \in \mathbf{F}_2^n} (-1)^{f(x)}(-1)^{u \cdot x}$$

where $u \cdot x$ denotes the usual scalar product. The Walsh coefficient $\widehat{\chi}_f(u)$ then estimates the Hamming distance between f and the affine function $u \cdot x + \varepsilon$, $\varepsilon \in \mathbf{F}_2$, both seen as Reed-Muller codewords [12]:

$$d_H(f, u \cdot x + \varepsilon) = 2^{n-1} - \frac{(-1)^\varepsilon}{2} \widehat{\chi}_f(u) \ .$$

It is well-known that a combining function must fulfill some criteria to yield a cryptographically secure combination generator (see e.g. [6]). For correlation attacks and fast correlation attacks the main criterion is that f should be far from all affine functions regarding Hamming distance. The existence of a good approximation of f by an affine function makes fast correlation attacks feasible [2, 8, 9]. The Hamming distance between f and the set of affine functions, called the *nonlinearity* of f, is given by

$$\mathcal{NL}(f) = 2^{n-1} - \frac{1}{2} \max_{u \in \mathbf{F}_2^n} |\widehat{\chi}_f(u)| \ .$$

Finally, the combination generator is vulnerable to correlation attacks [19] if the output of the combining function statistically depends on one of its inputs. More generally, Siegenthaler [18] introduced the following criterion:

Definition 1 *A Boolean function is* t-th order correlation-immune *if the probability distribution of its output is unaltered when any t input variables are fixed.*

This property equivalently asserts that the output of f is statistically independent of any linear combination of t input variables. The correlation-immunity order of a function can be characterized by its Walsh spectrum:

Proposition 2 [20] *A Boolean function f is t-th order correlation-immune (denoted $CI(t)$) if and only if*

$$\forall u \in \mathbf{F}_2^n,\ 1 \leq wt(u) \leq t,\ \ \widehat{\chi}_f(u) = 0\ .$$

If f is $CI(t)$ then f if $CI(k)$ for any $k \leq t$ as well. Then the correlation-immunity order of a function f the highest integer t such that f is $CI(t)$. Equivalently f is said $(t+1)$-th order correlated.

Note that the correlation-immunity order of a function with n variables can not exceed $(n-1)$. This comes from Parseval's relation:

$$\sum_{u \in \mathbf{F}_2^n} \left(\widehat{\chi}_f(u)\right)^2 = 2^{2n}\ .$$

This equality insures that there always exists a possibly high correlation order allowing correlation attack. It also points out the existence of a trade-off between the correlation-immunity order and the non-linearity of a function. The higher the correlation-immunity order is, the lower the nonlinearity (i.e. the correlation value) may be. The correlation-immunity order t of a Boolean function f with n variables also provides an upper bound on its degree [18]: $\deg(f) \leq n-t$. Moreover, if f is balanced, we have $\deg(f) \leq n-t-1$.

3 The Decimation Attack

3.1 Description of the Algorithm

To deal with a more realistic attack, we will consider only ciphertext attack. So this will limit the drawback of ciphertext length increasing. Then the BSC with error probability p will model at the same time the correlation of the Boolean function $P[x_t = \sigma] = p_\sigma$ and the bit probability of plaintext $p_0 = P[m_t = 0]$. We take $p_0 = 0.65$ which corresponds to most real-life cases. Then we have $p = p_0 + p_\sigma - 2.p_0 p_\sigma$.

Let be a target LFSR of length L such that there exists $d|2^L - 1$ satisfying Proposition 1. In all these cases (it has been confirmed by exhaustive computing using GMP library), the best d yields a simulated LFSR of length at most $\frac{L}{2}$ (for extended tables see [5]).

With the decimated ciphertext sequence $\rho = \sigma[d]$ we try to recover the corresponding output sequence y of the simulated LFSR which is the decimated sequence $x[d]$ of the (original) target LFSR output sequence. It simply consists of a Siegenthaler attack [19] on this simulated LFSR(see Figure 1). It then needs only 2^{L^*} exhaustive searches.

Any kept L^*-bit candidate is used to generate L bits of the decimated sequence $x[d]$. Since each bit of sequence $x[d]$ is a bit of sequence x as well, it can be described by two different equations. One has L^* variables (the bit is seen as a bit of sequence $x[d]$). The other has L variables (the bit is seen as a bit of sequence x). Example 2 illustrates that.

The system of equations in L variables has obviously rank L^*. Then by taking L^* principal variables, we can express them (gaussian elimination on the principal subsystem) depending on the L^* other variables taken as parameters. An additional L^*-bit exhaustive search on these parameters allows to retrieve the correct L-bit initialization.

Note that in all our experiments, the correct L^*-bit initialization of the simulated LFSR was always detected with the best or second best estimator value (first or second rank, see farther), thus insuring detection and limiting cost of additional exhaustive search. However, to prevent a non such optimal detection, first step of L^*-bit exhaustive search is conducted on a few shifted decimated sequences $\sigma_0[d], \sigma_1[d], \ldots, \sigma_k[d]$ where $\sigma_i[d]$ means that we decimate σ from index position i $(0 \leq i < d)$. Sorting always allowed to detect the correct L^*-bit initialization. Good experimental values are $2 \leq k \leq 8$ and always less than $k < L^*$.

We compute for each of the 2^{L^*} possible initializations the value of estimator

$$E = \sum_t (x_i^t[d] \oplus \sigma^t[d] \oplus 1)$$

For the correct L^*-bit initialization, E has Gaussian distribution with mean value $N.(1-p)$ and variance $\sigma^2 = N.p.(1-p)$ ($\mathcal{H}_1$ hypothesis) whilst for all the wrong ones E has Gaussian distribution with mean value $N.\frac{1}{2}$ and variance $\sigma^2 = N.\frac{1}{4}$ ($\mathcal{H}_0$) where N is the minimum number of required ciphertext bits of $\sigma[d]$.

To discriminate $\mathcal{H}_0$ from $\mathcal{H}_1$ we use a decision threshold $\mathcal{T}$. If $|E| > \mathcal{T}$ then $\mathcal{H}_1$ is accepted and the initialization is kept otherwise $\mathcal{H}_0$ is chosen and the initialization is rejected. The minimum number N of required ciphertext bits of $\sigma[d]$ depends on the number of wrong decisions that we accept. This number (as well as the threshold $\mathcal{T}$) is determined by the *false alarm probability* ($pfa = P[E > \mathcal{T}|\mathcal{H}_0]$) and the *non detection probability* ($pnd = P[E < \mathcal{T}|\mathcal{H}_1]$). If Φ denotes the normal distribution function

$$\Phi(x) = \frac{1}{sqrt 2\pi} \int_{\infty}^{x} \exp(\frac{-t^2}{2})\, dt$$

then we have [4]

$$pfa = 1 - \Phi\left(a = \frac{\mathcal{T} - \frac{N}{2}}{\sqrt{\frac{N}{4}}}\right) \qquad pnd = \Phi\left(b = \frac{\mathcal{T} - N(1-p)}{\sqrt{N(1-p)p}}\right)$$

Finally we obtain

$$N = \left(\frac{2b\sqrt{p(1-p)} - a}{1-2p)}\right)^2 \quad (1)$$

$$\mathcal{T} = \frac{1}{2}\left(a\sqrt{N} + N\right) \quad (2)$$

In terms of ciphertext we then need $N.d$ bits. Global complexity of the attack is then in $\mathcal{O}(2^{L^*+1})$ with no memory requirements.

3.2 Simulation Results

Now let us present simulation results of our attack.

Example 2 *CI(0) function.*
Let us consider a LFSR of length $L = 40$ *with feedback polynomial:*

$$\begin{aligned} P(x) &= 1 + x^2 + x^3 + x^4 + x^5 + x^6 + x^8 + x^{11} + x^{12} + x^{15} + x^{17} + x^{19} + \\ & x^{22} + x^{24} + x^{25} + x^{26} + x^{27} + x^{28} + x^{29} + x^{35} + x^{40} \end{aligned}$$

Consider a BSC with noise probability of $p = 0.49(P_0 = 0.65$ *and* $P[x_t = y_t] = 0.534)$ *modelling a CI(0) Boolean function and the plaintext noise at the same time. Since* $2^{40} - 1 = 3.5^2.11.17.31.41.61681$, *by choosing* $d = 1,048,577$ *as decimation factor for the LFSR output sequence* σ, *we obtain a simulated shorter LFSR of length* $L^* = 20$ *with feedback polynomial:*

$$P^*(x) = 1 + x + x^2 + x^3 + x^6 + x^7 + x^8 + x^{11} + x^{12} + x^{13} + x^{15} + x^{18} + x^{20}$$

With $pnd = 0.1$ *and* $pfa = 0.1$, *we then need by Equation 2,* $N = 13050$ *bits of decimated sequence* $\rho = \sigma[d]$ *that is to say* $N.d = 2^{33}$ *bits of ciphertext. Complete computation time (*i.e. *recovering the correct inital state, with* $k = 2$*) required about 15 minutes on PII 400 Mhz with less than 1 Mo of memory.*

Note that the 20,971,541st bit (e.g.), when seen as belonging to sequence x *is described by the following (40 variables) equation (where* x_i $0 \leq i \leq 39$ *denotes ith unknown bit of the L-bit initialization):*

$$\begin{aligned} b_{20,971,541} &= x_2 + x_4 + x_6 + x_7 + x_{14} + x_{16} + x_{17} + x_{19} + x_{20} + x_{22} + x_{23} \\ & +x_{25} + x_{26} + x_{28} + x_{29} + x_{30} + x_{33} + x_{36} + x_{37} \end{aligned}$$

whilst, when seen as 21st bit of $x[d]$, *we have the following (20 variables) equation (where* y_i $0 \leq i \leq 19$ *denotes ith unknown bit of the* L^**-bit initialization):*

$$b_{20,971,541} = y_0 + y_1 + y_2 + y_3 + y_6 + y_7 + y_8 + y_{11} + y_{12} + y_{13} + y_{15} + y_{18}$$

Example 3 *CI(1) function.*
Consider now two LFSRs P_1 *of length* $L_1 = 34$ *and* P_2 *of length* $L_2 = 39$ *combined by a CI(1) Boolean function.*

$$\begin{aligned} P_1(x) &= 1 + x^4 + x^5 + x^6 + x^7 + x^{12} + x^{13} + x^{14} + x^{15} + x^{16} + x^{18} + \\ & x^{20} + x^{21} + x^{22} + x^{24} + x^{25} + x^{29} + x^{30} + x^{31} + x^{33} + x^{34} \\ P_2(x) &= 1 + x + x^2 + x^3 + x^4 + x^5 + x^{14} + x^{15} + x^{16} + x^{17} + x^{19} + x^{21} + \\ & x^{24} + x^{21} + x^{24} + x^{27} + x^{28} + x^{29} + x^{31} + x^{32} + x^{36} + x^{37} + x^{39} \end{aligned}$$

We take the same BSC model as Example 2 but for a CI(1) Boolean function. This implies we must try all the possible initial states of at least two LFSRs at the same time thus requiring with previous known attacks to consider a unique *target LFSR of length 73. The best decimation factor in this case is* $d = 131,073$ *which divides* $2^{34} - 1$*. So we obtain these two new simulated LFSRs:*

$$\begin{aligned} P_1^*(x) &= 1 + x + x^2 + x^5 + x^6 + x^7 + x^9 + x^{10} + x^{11} + x^{12} + x^{13} + x^{15} + \\ & \quad x^{17} \\ P_2^*(x) &= 1 + x + x^4 + x^6 + x^7 + x^{14} + x^{16} + x^{17} + x^{18} + x^{20} + x^{22} + \\ & \quad x^{24} + x^{28} + x^{30} + x^{32} + x^{34} + x^{39} \end{aligned}$$

With $pnd = 0.1$ *and* $pfa = 0.1$*, we then need by Equation 2,* $N = 13050$ *bits of decimated sequence* $\rho = \sigma[d]$ *that is to say* $N.d = 2^{30}$ *bits of ciphertext. Complexity is in* $\mathcal{O}(2^{56})$ *with less than 1 Mo of memory. Partial experiment (*$k = 8$*) have been conducted on a PII 400 Mhz to verify that final* pfa *was effectively close to zero.*

Note that rank of the system in L variables is 56, thus with 17 parameters. Then the additional exhaustive search is on 17 bits and is negligible compared to the first exhaustive search step.

3.3 Decimation Attack Resistance Criterion

By direct use of Proposition 1, we can define the following criterion for Decimation Attack resistance.

Proposition 3 *Let* $L \in \mathbb{N}$ *Any feedback polynomial of degree L will resist the decimation attack if and only if* $\forall d < 2^L - 1$ *such that* $d|(2^L - 1)$*, the multiplicative order of 2 in* $\mathbb{Z}_{T^*}$ *is equal to L where* $T^* = \frac{2^L-1}{\gcd(2^L-1,d)}$

We then obviously have

Corollary 1 *If* $T = 2^L - 1$ *is prime then any feedback polynomial of length L will resist the decimation attack.*

In fact, finite field theory allows to precise im a simpler way this criterion.

Theorem 1 *(Subfield Criterion) [10] Let* $\mathbb{F}_q$ *be a finite field with* $q = p^n$ *elements. Then every subfield of* $\mathbb{F}_q$ *has order* p^m*, where* m *is a positive divisor of* n*. Conversely, if* m *is a positive divisor of* n*, then there is exactly one subfield of* $\mathbb{F}_q$ *with* p^m *elements.*

Then we can easily state the resistance criterion as follows:

Proposition 4 *Any feedback polynomial of length L, such that L is prime, will resist the decimation attack.*

Proof.
straightforward by direct application of Theorem 1 and Proposition 3. □

Equivalently, resistance is ensured as soon as the field $\mathbb{F}_{2^L}$ contains no subfield. It becomes then obvious that, when decimation attack is possible, there exists a value of L^* at most equal to $\frac{L}{2}$.

Note that this criterion is very new and must not be mistaken with that of **relative primality** of periods when considering the sum of outputs of LFSRs as described by the following theorem:

Theorem 2 *[10, p. 224] For each $i = 1, 2, \dots, h$, let σ_i be an ultimately periodic sequence in $\mathbb{F}_q$ with least period r_i. If $r_1, r_2, \dots, r_h$ are pairwise relatively prime, then the least period of the sum $\sigma_1 + \sigma_2 + \cdots + \sigma_h$ is equal to product $r_1 r_2 \dots r_h$.*

When designing a stream cipher system, one must carefully check the degree of the chosen feedback polynomials (in connection with correlation order) to resist the Decimation Attack. Table 1 gives some values of L satisfying this new resistance criterion ("prime" relates to Corollary 1 and "order" relates to Proposition 3).

Comment	L
Prime	5 - 7 - 13 - 17 - 19 - 31 - 61 - 89 - 107 - 127
Order	11 - 23 - 29 - 37 - 41 - 43 - 47 - 53 - 59 - 67 - 71 - 73 83 - 97 - 101 - 109 - 113 - 131 - 139 - 149 - 151 - 157 - 163 167 - 173 - 178 - 179 - 181 - 191 - 193 - 197 - 199 211 - 223 - 227 - 229 - 233 - 239 - 241 - 251

Table 1: Values of $L < 256$ resisting the Decimation Attack

A complete list of parameters d, T^* and L^* up to $L = 256$ has been computed and can be found in [5]

4 Comparison with the Best Known Attacks

Canteaut and Trabbia (CT) in [2] recently obtained the best known attack on stream ciphers. They considered parity-check equations of weight $\delta = 4, 5$ with Gallager iterative decoding. However, the main drawback of their approach, despite its relative efficiency, remains the huge amount of required memory both for preprocessing (generation of the parity-check equations) and decoding steps. This latter, though being the best, still requires higher complexity than suitable for frequent key recovering.

We now give here comparison of our attack (DA)(when possible, see Section 3.3) with CT attack. Suppose that by applying condition of Proposition 1 a reduction of ΔL bits on exhaustive search is possible. Then our attack has complexity $\mathcal{C}_{DA} = \mathcal{O}(2^{L-\Delta L})$ and a negligible amount of memory. Let us now compute the complexity gain from CT attack. In [2] the complexity is given by the following formula:

$$\mathcal{C}_{CT} = 5.(\delta - 1).\frac{K_\delta}{C_{\delta-2}(p)}.2^{\alpha_\delta(p) + \frac{L}{\delta - 1}}$$

where δ is the weight of the parity-check equation ($3 \leq \delta \leq 5$) and $C_{\delta-2}(p)$ is the BSC capacity (with overall error probability $p_{\delta-2} = \frac{1}{2}(1-(1-2p)^{\delta-2})$) *i.e.* $C_{\delta-2}(p) = 1 + p_{\delta-2}\log(p_{\delta-2}) + (1-p_{\delta-2})\log(1-p_{\delta-2})$. The value $K_\delta \approx 1$ if $\delta \geq 4$ and $K_3 \approx 2$. Finally $\alpha_\delta(p) = \frac{1}{\delta-1}\log_2[(\delta-1)!\frac{K_\delta}{C_{\delta-2}(p)}]$.

The complexity gain of our attack is then

$$\frac{\mathcal{C}_{CT}}{\mathcal{C}_{DA}} = 5.(\delta-1).\frac{K_\delta}{C_{\delta-2}(p)}.2^{\alpha_\delta(p)+\frac{L}{\delta-1}-\Delta L}$$

It is easy to see that the gain increases with δ. Now using higher weight precisely constitutes the central point of CT attack. That particularly reinforces the impact of our technique. Moreover, the gain increases with the error probability p. This fact ensures greater efficiency than with CT attack in real-life cases.

Concerning the memory requirement, CT attack needs $(\delta-1).\frac{K_\delta}{C_{\delta-2}(p)} + 2^{\alpha_\delta(p)+\frac{L}{\delta-1}+1}$ computer words of memory. This once again increases with δ and p.

To illustrate this gain Tables 2 and 3 compare our simulation results (see Section 3.2) with those of CT attack [1, 2]. Yet requiring a longer sequence,

	Decimation Attack	CT Attack ($\delta = 5$)
Ciphertext bits	2^{33}	2^{20}
Complexity	2^{21}	2^{59}
Memory (bytes)	< 1024	2^{38}

Table 2: Comparaison on Example 2

we have a complexity gain of 2^{39} with no memory requirement. However CT attack still presents advantages in terms of ciphertext length only for relatively short LFSRs. This is no longer the case as soon as L increases as we can see it for Example 3 ("ppm" means preprocessing with memory and "ppwm" means preprocessing without memory).

	DA	CT ($\delta = 5$) (decoding step)	CT ($\delta = 5$) (ppm step)	CT ($\delta = 5$) (ppwm step)
Ciphertext bits	2^{30}	2^{29}	-	-
Complexity	2^{56}	2^{67}	2^{56}	2^{81}
Mem. (bytes)	< 1024	2^{39}	2^{57}	-

Table 3: Comparaison on Example 3

Note that preprocessing step in CT attack must be done once and for all but for each different feedback polynomial of length L. On the contrary, Decimation Attack apply to all polynomials of degree L (for suitable L) and no preprocessing is required.

5 Conclusion

A new attack on stream ciphers, called *Decimation Attack* has been presented which overperformed all the previous known attacks, provided that a given new criterion is satisfied. Particularly, Decimation Attack is not possible when L is prime, where L is the length of the feedback polynomial. Only very few values of L allow to resist this attack.

References

[1] Anne Canteaut, Email Communication, March 2000.

[2] A. Canteaut, M. Trabbia, *Improved Fast Correlation Attacks using Parity-check Equations of weight 4 and 5.* Eurocrypt'2000, Bruges.

[3] V. Chepyzhov, B. Smeets, *On a Fast Correlation Attack on Stream Ciphers*, Advances in Cryptology - EUROCRYPT'91, LNCS 547, Springer Verlag, 1991.

[4] W. Feller *An Introduction to Probability Theory* , Wiley, 1966.

[5] http://www-rocq.inria.fr/codes/Eric.Filiol/index.html

[6] E. Filiol, C. Fontaine, *Highly Nonlinear Balanced Boolean Functions with a Good Correlation-Immunity*, Advances in Cryptology - EUROCRYPT'98, LNCS 1403, Springer Verlag, 1998.

[7] R.G. Gallager, *Low-density parity-check codes*, IRE Trans. Inform. Theory, IT-8:21-28, 1962.

[8] T. Johansson, F. Jönsson, *Improved Fast Correlation Attack on stream Ciphers via Convolutional Codes*, Advances in Cryptology - EUROCRYPT'99, LNCS 1592, pp 347–362, Springer Verlag, 1999

[9] T. Johansson, F. Jönsson, *Fast Correlation Attack based on Turbo Codes Techniques*, Advances in Cryptology - EUROCRYPT'99, LNCS 1592, pp 347–362, Springer Verlag, 1999

[10] R. Lidl, H. Niederreiter *Introduction to Finite Fields and their Applications*, Cambridge University Press, 1994.

[11] J.L. Massey *Shift-Register Synthesis and BCH Decoding*, IEEE Trans. on Info. Theory, Vol. IT-15, Jan. 1969.

[12] F. J. MacWilliams, N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland Mathematical Library, North-Holland, 1977.

[13] W. Meier, O. Staffelbach, *Fast Correlation Attack on certain Stream Ciphers*, J. of Cryptology, pp 159–176, 1989.

[14] M. Mihaljevic, J. Dj. Golic, *A Fast Iterative Algorithm for a Shift-Register Initial State Reconstruction given the Noisy Output Sequence*, Proc. Auscrypt'90, LNCS 453, Springer Verlag, 1990.

[15] W. Penzhorn, *Correlation Attacks on Stream Ciphers: Computing low Weight Parity Checks based on Error-Correcting Codes*, FSE'96, LNCS 1039, Springer Verlag, 1996.

[16] R.A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer Verlag, 1986.

[17] E.S. Selmer, *Linear Recurrence Relation over Finite Fields*, Ph. D Thesis, University of Bergen, Norway, 1966.

[18] T. Siegenthaler, *Correlation Immunity of Nonlinear Combining functions for Cryptographic Applications*, IEEE Transactions on Information Theory, Vol. 35 Nr 5, September 1984, pp 776–780.

[19] T. Siegenthaler, *Decrypting a Class of Stream Ciphers using Ciphertext Only*, IEEE Transactions on Computers, C-34, 1, pp 81–84, 1985.

[20] G. Xiao, J. Massey, *A Spectral Characterization of Correlation Immune Functions*, IEEE Transactions on Information Theory, Vol 34, pp 569 – 571, may 1988.