Journal of the Indian Statistical Association
Vol.42, 1, 2004, 1-34

Invited Paper

SOME STATISTICAL ATTACKS ON STREAM CIPHER CRYPTOSYSTEMS

Bimal Roy and Sarbani Palit
Indian Statistical Institute, Kolkata

Abstract

Stream Cipher models are cryptanalysed using statistical techniques assuming that the detailed architecture of the model (except for the key) and ciphertexts are available. The idea is to estimate the secret key with a "reasonable" computational complexity. The methodology used involves statistical testing of hypothesis, maximum likelihood estimation, Markov chain etc.

Key Words : *Stream ciphers, cryptanalysis, correlation attack, fast correlation attack.*

1 Introduction

Cryptography is the science or art of secret writing. A general cryptographic system consists of

a) P, a finite space of plain texts or messages.

b) C, a finite space of ciphertexts.

*Received (revised version) : May, 2004

c) K, a finite set of keys.

d) An encryption function $E : P \times K \rightarrow C$ such that each function for a given $k \in K$, the restriction $e_k : P \rightarrow C$ defined by $e_k(m) = E(m, k)$ is invertible, i.e., there exists a function $d_k : C \rightarrow P$ such that $d_k(e_k(m)) = m$ for every $m \in P$.

Also it is assumed that d_k is easily computable, but is computationally hard if k is unknown.

Cryptanalysis (popularly known as code breaking) is the other side of the coin. It is assumed that ciphertexts are always available to an attacker and in some cases, same plaintext may also be available. In this paper, "ciper text only attack" is considered where the ciphertexts are available along with knowledge about P, C, K and the functional form of E. Cryptanalysis attempts to recover the plain text/ secret key. Even a partial recovery is generally regarded as success for the cryptanalysts.

In this paper, "ciper text only attack" on a widely used cryptosystem, called stream ciphers, is considered. The attacks are mostly statistical in nature and thus demands the attention of the statisticians. The tools used are : maximum likelihood estimation, testing of hypothesis, Markov chain etc.

In Section 2, the general stream cipher model is described in details. The first statistical attack on this model was due to Siegenthaler ([6], [24]) which was subsequently modified/extended by the authors. Section 3 details these. In this attack the combining Boolean function was assumed to be known. Assuming that the function was unknown, authors proposed a method of estimating the function. This is described in Section 4. Meier and Staffelbach [9] proposed "fast correlation attack" which had significantly lower complexity compared to the previous attacks. There algorithm was then slightly improved by the authors. These constitute Section 5. Two other attacks due to Mihaljevic and Golic ([14], [15]) and Filiol [23] are outlined in Section 6 and 7 respectively. Section 8 deals with Boolean functions with memory and are due to Palit and Dasgupta [27].

2 The stream cipher system architecture and its components

Figure 1 shows the general form of a popular stream cipher system. The generator G produces a random sequence called the 'keystream' (Y). This is X-ORed (added modulo 2), bit-by-bit with the encoded message called the 'plaintext' (M) to produce the 'ciphertext' (C). For decryption, the same keystream must be X-ORed with the ciphertext (in synchronization with the encryption process) to retrieve the encoded plaintext.

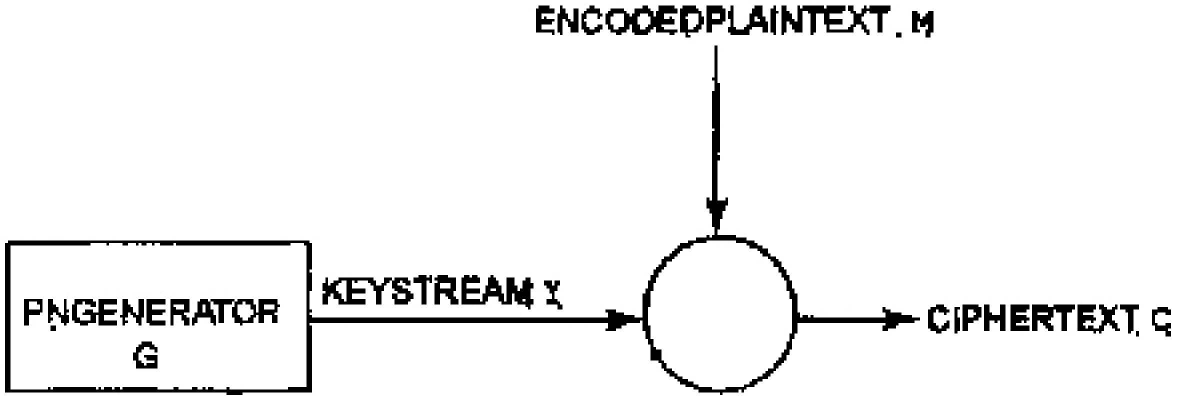

Figure 1 : Block diagram of a stream cipher system.

A cryptosystem is said to have *perfect secrecy* if $p(x|y) = p(x)$ $\forall$ $x \in P, y \in C$. P is a finite set of possible plaintexts, C is a finite set of possible ciphertexts and K stands for the set of possible keys. This means that the *a posteriori* probability that the plaintext is x, given that the ciphertext y is observed, is identical to the *a priori* probability that the plaintext is x. Let $e_K \in E$ be the encryption rule and $d_K \in D$ be the decryption rule. Then Shannon [1] provides another characterization of perfect secrecy : Suppose P, K, C, E, D represents a cryptosystem with $|P| = |K| = |C|$. Then the cryptosystem provides perfect secrecy if and only if every key is used with uniform probability $\frac{1}{|K|}$ and for every $x \in P$ and every $y \in C$ there is a unique key such that $e_K(x) = y$. A well known realization of a perfectly secret system is the Vernam One-time pad. This consists of bit-by-bit X-ORing of the plaintext and keystream to obtain the ciphertext. Decryption is performed by X-ORing the ciphertext and keystream. Most importantly, each key must be used only once which makes the system unconditionally secure and must be of length at least that of the plaintext.

One-time pads, therefore, have two major disadvantages– large length requirements and also the necessity of not ever repeating. A perfectly random sequence which would never repeat is impossible to obtain in practice. Hence, random sequences must be replaced by actually available pseudo-noise (pn) sequences. These sequences are required to satisfy some standard notions of randomness such as Golomb's randomness postulates [2].

A linear feedback shift register (LFSR) is commonly used to implement a pn sequence. It is both efficient and easy to implement in hardware as well as software. The nth bit of the output generated serially by an LFSR of length d is related to the previous d bits by the linear equation

$$x_i = a_1 x_{i-1} + a_2 x_{i-2} + \cdots + a_d x_{i-d}, \tag{1}$$

$a_1, \ldots, a_d$ being binary constants which, along with the d initial values, characterize the pn sequence. The above equation is often described by means of the polynomial $a(X) = 1 + a_1 X + a_2 X^2 + \cdots + a_d X^d$, known as the feedback or connection polynomial. When the feedback polynomial is *primitive*, i.e., cannot be factorized and any root of it generates the entire field, the period (cycle-length, after which repetition sets in) of the sequence generated is of *maximal length* and equals $2^d - 1$. The longer the period length of the sequence, more is the 'pseudorandomness' of the sequence. Having a long period length is of vital importance to the security of the cryptosystem.

An example of an LFSR with a primitive feedback polynomial $1 + x + x^4$ is shown below in Figure 2. The period of the sequence is $2^4 - 1 = 15$.

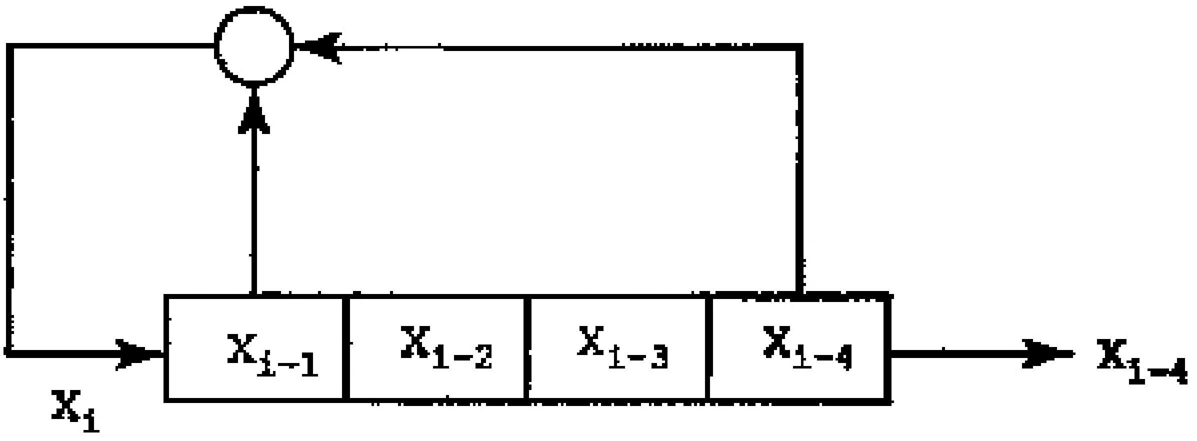

Figure 2 : Block diagram of a linear feedback shift register (LFSR) of lenght 4.

Since the bits of the LFSR sequence satisfy a linear recurrence relationship, the use of such a sequence as the keystream leads to an attack of by the Berlekamp-Massey shift register synthesis algorithm [4]. In order to eliminate the possibility of attacks along these lines, the outputs of several LFSRs are combined using a nonlinear Boolean function in order to destroy the inherent linearity present in the keystream. The corresponding system which is one of the most popular stream cipher systems, is shown in Figure 3. The system is initialized with a set of initial conditions for the LFSRs which is the secret *key*.

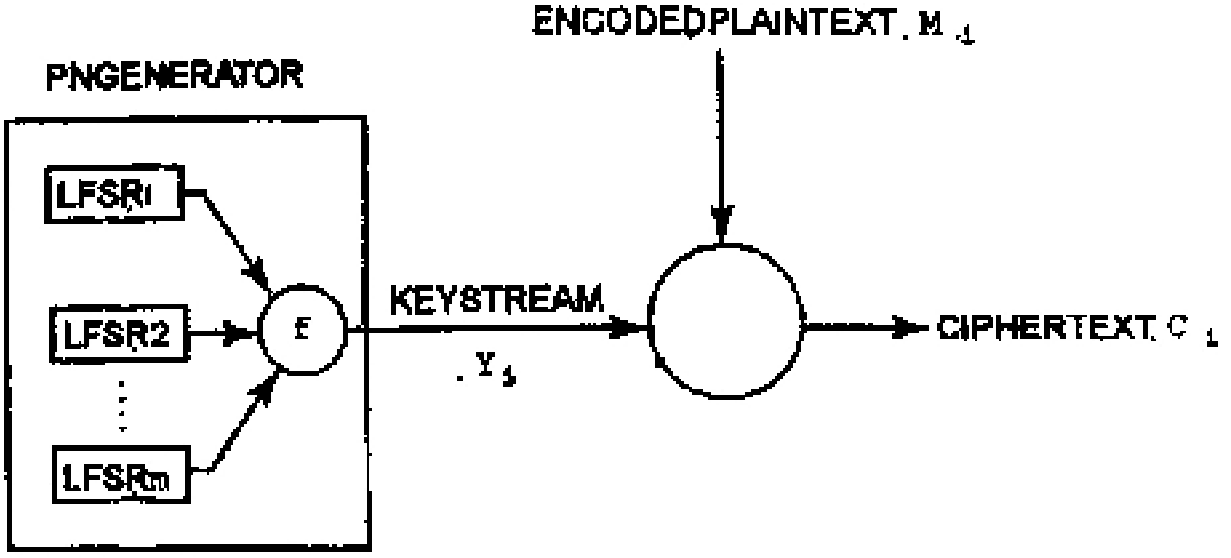

Figure 3. A stream cipher system driven by LFSRs with a combining function.

In such a system, the attacker may face one or more of the following problems, *viz.* unknown initial conditions of the LFSRs, unknown LFSR polynomials, unknown combining function, availability of limited cipherlength and the need for computation in a reasonable amount of time.

An LFSR of length d_j, has $2^{d_j} - 1$ different choices of the initial conditions. The total number of LFSR initial conditions possible for the system shown in Figure 3 is

$$K = \prod_{j=1}^{m} (2^{d_j} - 1) \quad \text{(the all zero condition is never used)}$$

If the feedback polynomials of the shift registers are unknown, and R_j is the number of (possibly primitive) feedback polynomials for the jth LFSR then

$$K = \prod_{j=1}^{m} R_j (2^{d_j - 1}).$$

Note that $R_j = \frac{1}{d}\phi(2^{d_j} - 1)$ where $\phi(p)$ is the *Euler phi-function* and gives the number of integers from the set $\{0, 1, \cdots, p-1\}$ that are relatively prime to p [2]. Hence, in a situation when the attacker has access to the ciphertext only, for a *brute force* attack, he must attempt the decryption using all K keys and wait till a meaningful decrypted message is obtained. This is computationally infeasible for LFSRs of even moderate sizes.

3 Siegenthaler's correlation attack

Because of the fact that the probability of 0 in the coded plain text is not exactly equal to half, there is often a non-zero correlation between the keystream and each of the LFSR output sequences. This correlation was first exploited by Siegenthaler [6] to form a *divide and conquer approach* so that the correct initial condition (i.c.) of each LFSR may be determined independently of the i.c.s of the others. He assumed that the shift register sizes and the form of the nonlinear combining function are known. The number of trials to find the i.c.s is then dramatically reduced to

$$\sum_{j=1}^{m} R_j 2^{d_j - 1}$$

3.1 The method

Let N denote the cipherlength available, $X_1^j, \ldots, X_N^j$ the sequence produced by the jth LFSR, and d_j, the size of the jth LFSR, $j = 1, \ldots, m$. Since each of the LFSR outputs are *pn* sequences, X_i^j, $i = 1, \cdots, N$ are i.i.d random variables with

$$P_X(X_i^j = 0) = P_X(X_i^j = 1)$$

for all i and j. Further, if the nonlinear combining function f is *balanced, i.e. its output has an equal number of zeros and ones,* then

$$P(Y_i = 0) = P(Y_i = 1).$$

Let $p_0 = P(M_i = 0)$ and $p_j = P(C_i = X_i^j)$. Even though p_j is not directly related to the correlation between C_i and X_i^j, an attack based on the deviation of this probability from one-half is conventionally referred to as a 'correlation attack'. Note that

$$p_j = P(Y_i = X_i^j | M_i = 0)P(M_i = 0) + P(Y_i \neq X_i^j | M_i = 1)P(M_i = 1) \tag{1}$$

Since Y_i and X_i^j are both independent of M_i, we have the simplification

$$p_j = q_j p_0 + (1 - q_j)(1 - p_0) \tag{2}$$

where $q_j = P(Y_i = X_i^j)$.

Consider the random sequence

$$Z_i^j = \begin{cases} 1 & \text{if } C_i = X_i^j, \\ 0 & \text{if } C_i \neq X_i^j. \end{cases}$$

It can be deduced that Z_i^j is a Bernoulli random variable and $\sum_{i=1}^{N}(1 - Z_i^j) \sim Bin(N, 1 - p_j)$. Thus, for large N, the empirical measure of concurrence between C_i and X_i^j given by

$$\alpha_j = N - 2\sum_{i=1}^{N}(1 - Z_i^j), \quad 1 \le j \le m \tag{3}$$

is approximately normally distributed with mean and variance

$$\begin{aligned} m_{\alpha_j} &= N(2p_j - 1), \\ \sigma^2_{\alpha_j} &= 4Np_j(1 - p_j). \end{aligned} \tag{4}$$

For all known codes of the plaintext, the probability p_0 is generally different from 0.5. Hence, $p_j = 0.5$ if and only if $q_j = 0.5$. This happens to be the case when the combining function is first order correlation immune i.e., if the function is $f(X_1, X_2, \ldots, X_n)$, then $P(f = X_i) = \frac{1}{2}, \forall i = 1, \ldots, n$. We shall not discuss the attacks for correlation immune functions; the reader is referred to Palit and Roy [7] for information on strategies in this case. If p_j is different from 0.5, then α_j is different from 0. On the other hand, if an arbitrary trial sequence is used in place of X_i^j, then $p_j = 0.5$, and consequently

$$\begin{aligned} m_{\alpha_j} &= 0, \\ \sigma^2_{\alpha_j} &= N. \end{aligned} \tag{5}$$

Thus, the question of determining the correctness of a candidate i.c. of the jth LFSR reduces to a test of the null hypothesis $H_0 : m_{\alpha_j} = 0$ against the alternative hypothesis $H_1 : m_{\alpha_j} \neq 0$. Note that the sequence $X_1^j, \ldots, X_N^j$ needed for computing the test statistic, α_j, is uniquely determined by the candidate i.c., once the connection polynomial is known.

Let us assume, without loss of generality, that $p_j > 0$ for a particular i.c. of the jth LFSR. If the cut-off used for the test statistic α_j is T, then the probability of false alarm is

$$P(\alpha > T|H_0) = 1 - \Phi(T/\sqrt{N}),$$

while the probability of miss is

$$P(\alpha \leq T|H_1) = \Phi((T - (N(2p_j - 1)))/2\sqrt{Np_j(1 - p_j)}),$$

where $\Phi(.)$ is the standard Normal distribution function. Siegenthaler ([6], [24]) recommends setting the threshold T to ensure a predetermined maximum probability of miss. If several candidate i.c.s exceed the threshold, then all of these should be used to try and decode the ciphertext. If no candidate i.c. is found to exceed the threshold, then a different connection polynomial may be tried out.

3.2 A modification of Siegenthaler's approach

The above approach can fail in two ways: (a) the correct initial condition may be missed, or (b) there may be too many false alarms. The twin objective of the decision-making process is to restrict the chances of both types of failures. However, a precise definition of success is necessary in order to examine the feasibility of breaking a code with a given cipherlength. A reasonable approach would be to define 'success' as the situation when the correct initial condition belongs to the selected list of solutions along with k wrong initial conditions. The number k has to be chosen before any analysis of performance. A high value of k would mean that a large number of candidate solutions have to be examined by actually generating the 'deciphered' texts – a prospect which can hardly be described as 'success'. Thus, k has to be a reasonably small number. The choice $k = 0$ was made in [7] and implicitly by Roy [8]. Of course, all the calculations can be generalized for $k > 0$. This choice of k implies that 'success' is defined as the case when the shortlist of selected initial conditions contains only the correct one.

Let us assume that $p_0 > 0.5$. Let f_{ic} be the observed fraction of coincidences, $\sum Z_i/N$, when the chosen initial condition is correct. Suppose that f_{iw} is the largest value of $\sum Z_i/N$ when a wrong initial condition is used. Siegenthaler's method would be successful (in the sense described above) if

$$f_{iw} \leq (T/N + 1)/2 < f_{ic}.$$

However, f_{ic} can be larger than f_{iw} even if both of these are on the same side of the threshold. A correct determination of the initial condition is possible in such a case, by modifying Siegenthaler's approach as follows. Let f_i be the fraction of coincidences between the ciphertext and the ith input. One may check for the maximum of f_i over all possible initial conditions. In the modified approach, the maximizer is identified as the correct initial condition of the ith input.

If $p_0 < 0.5$, the minimizer of f_i over all possible initial conditions should be identified as the correct initial condition.

A theoretical analysis [7] showed that this modified algorithm requires a smaller cipherlength for correct determination of the initial conditions.

4 Estimation of the combining function

Let us now consider the situation when the combining function is unknown and non-correlation immune. We initially assume that the LFSR i.c.'s have been correctly identified. Identifying the combining function amounts to determining the 2^m binary output values in the corresponding truth table. We treat these numbers, denoted here by $y_0, y_1, \ldots, y_{2^m-1}$, as unknown parameters. These parameters control the distribution of the cipher stream. Using the knowledge of the inputs and the cipherstream, we proceed to obtain the maximum likelihood estimate of these parameters.

As mentioned earlier, for all practical coding schemes, $p_0 \neq 0.5$. Given that $Y_i = y_j$, the ith bit C_i of the ciphertext has the following probability mass function:

$$C_i = \begin{cases} 1 & \text{with probability } 1 - p_0 + (2p_0 - 1)y_j, \\ 0 & \text{with probability } p_0 - (2p_0 - 1)y_j, \end{cases}$$

This can be written in a more compact form as

$$P(C_i = c | Y_i = y_j) = [1 - p_0 + (2p_0 - 1)y_j]^c \cdot [p_0 - (2p_0 - 1)y_j]^{1-c}, \quad c = 0, 1.$$

Let $I_0, I_1, \ldots, I_{2^m-1}$ be the sets of indices of the bitstream that correspond to the 2^m different input combinations, respectively. [Note that these input combinations correspond to the outputs $y_0, y_1, \ldots, y_{2^m-1}$, respectively.) The sizes of these sets, $N_0, N_1, \ldots, N_{2^m-1}$, have a multinomial probability distribution with equal probabilities for each of the 2^m cells. The joint distribution of the ciphertext given the input streams, assuming independence of the bits of the ciphertext, is

$$\begin{aligned}&\prod_{j=0}^{2^m-1}\prod_{i\in I_j} P(C_i = c|Y_i = y_j)\\ &= \prod_{j=0}^{2^m-1}\prod_{i\in I_j}[1-p_0+(2p_0-1)y_j]^{C_i}\cdot[p_0-(2p_0-1)y_j]^{1-C_i}\end{aligned}$$

Thus, the likelihood of $y_0, y_1, \ldots, y_{2^m-1}$ given the input streams and the ciphertext is

$$\ell(y_0, y_1, \ldots, y_{2^m-1}) = \prod_{j=0}^{2^m-1}\prod_{i\in I_j}[1-p_0+(2p_0-1)y_j]^{C_i}\cdot[p_0-(2p_0-1)y_j]^{1-C_i}.$$

It may be noted that the parts that depend on each y_j appear as factors of the overall likelihood. Thus, we can work with one 'likelihood function' for each y_j, $j = 0, 1, \ldots, 2^m - 1$:

$$\ell_j(y_j) = \prod_{i\in I_j}[1-p_0+(2p_0-1)y_j]^{C_i}\cdot[p_0-(2p_0-1)y_j]^{1-C_i}.$$

Therefore, the MLE of y_j is

$$\widehat{y}_j = \begin{cases} 0 & \text{if } \ell_j(0)/\ell_j(1) > 1\\ 1 & \text{otherwise}\end{cases}$$

The condition $\ell_j(0)/\ell_j(1) > 1$ reduces to $[(1-p_0)/p_0]^{\sum_{i\in I_j}(2C_i-1)} > 1$. When $p_0 > .5$, as is the case for the Murray code, this further simplifies to $\sum_{i\in I_j} C_i < N_j/2$.

In summary, the MLE of y_j is

$$\widehat{y}_j = \begin{cases} 0 & \text{if } \sum_{i\in I_j} C_i < N_j/2,\\ \\ 1 & \text{if } \sum_{i\in I_j} C_i > N_j/2,\end{cases} \qquad j = 0, 1, \ldots, 2^m - 1.$$

In the unlikely event when $\sum_{i \in I_j} C_i = N_j/2$, both 0 and 1 are MLE, and one can assign one or the other without loss of generality.

It is shown in [7] that the cipherlength requirement can be obtained by setting a specified value to the probability

P(The entire m-input truth table is correctly identified)

$$\approx \left\{ \sum_{k=0}^{N} \left(\binom{N}{k} \left(\frac{1}{2^m}\right)^k \left(1 - \frac{1}{2^m}\right)^{N-k} \left(\sum_{l=[k/2]}^{k} \binom{k}{l} p_0^l (1-p_0)^{k-l} \right) \right) \right\}^{2^m} \tag{7}$$

When the initial conditions as well as the combining function are unknown, we can use the i.c. which maximizes $|\alpha_j|$ as the correct i.c. for the jth LFSR, and proceed with the above procedure for estimation f the combining function. It can be shown that the probability (7) remains unchanged, and is much larger in comparison to

P(All i.c.s are correctly identified)

$$= \prod_{j=1}^{m} \left[\sum_{y=0}^{N} \binom{N}{y} p_j^y (1-p_j)^{N-y} \left\{ 2\Phi\left(\frac{|N-2y|}{\sqrt{N}} \right) - 1 \right\}^{2^{d_j}-2} \right]. \tag{8}$$

Thus, the additional cipherlength needed for correct decryption in the absence of knowledge of the combining function is only marginal.

5 The fast correlation attack of Meier and Staffelbach

The correlation attacks described so far are based on carrying out an exhaustive search over possible initial conditions. Fast correlation attacks, however, attempt to reconstruct the entire LFSR sequence in an iterative fashion. This section presents the first algorithm of this kind, proposed by Meier and Staffelbach [9].

5.1 The basic theory

As in the last section we assume that the LFSR sequence is given by (1). The stream cipher system is viewed as a binary noisy channel with the LFSR output at its input. Its output is the ciphertext. The analysis is performed for a single LFSR, though a number of LFSRs can be analyzed similarly. (Consequently we drop the index j.) The channel is assumed to be such that

$$p = P(C_i = X_i) > 0.5 \tag{9}$$

We consider only Boolean functions implying that the coefficients of the polynomial are either 0 or 1. The number of non-zero coefficients a_l, $l = 1, 2, \ldots, d$ give the number of taps or feedback connections. We assume the existence of t such taps. Then, (1) can be rewritten as:

$$\sum_{0 \leq l \leq d, a_l = 1} X_{i-l} = 0 \tag{10}$$

having $t + 1$ terms. Note that a particular bit, say X_i can be placed in any of the $t + 1$ positions of (1). This implies that X_i simultaneously satisfies $t + 1$ equations of the form (1) or (2). Another important observation is that polynomial multiples of $a(X)$ generate linear relationships satisfied by X and in particular, powers of the form $a(X)^j$, $j = 2^i$, $\quad i = 1, 2, \cdots$, for which $a(X)^j = a(X^j)$. Thus, by repeated shifting of the sequence and 'squaring' of the polynomial, a large number of linear relations with the same number of taps are generated, all of which are satisfied by the bit X_i.

For example, consider the polynomial $1 + x + x^4$ and assume that the cipherlength available is $N = 65$. Then, listed below are the resulting polynomials produced by raising $1 + x + x^4$ to j, where $j = 2^i$, $\quad i = 1, 2, 3, \cdots$.

$$(1+x+x^4)^2$$
$$= 1+x^2+x^8+2x+2x^5+2x4(mod2) = 1+x^2+x^8$$
$$(1+x+x^4)^4$$
$$= 1+x^4+x^{16}+2x^2+2x^{10}+2x^8(mod2) = 1+x^4+x^{16}$$
$$(1+x+x^4)^8$$
$$= 1+x^4+x^{16}+2x^2+2x^{10}+2x^8(mod2) = 1+x^4+x^{16}$$
$$(1+x+x^4)^{16}$$
$$= 1+x^8+x^{32}+2x^8+2x^{40}+2x^{32}(mod2) = 1+x^8+x^{32}$$
$$(1+x+x^4)^{32}$$
$$= 1+x^{16}+x^{64}+2x^{16}+2x^{80}+2x^{64}(mod2) = 1+x^{16}+x^{64}$$

Note that the order of the last polynomial is 64 *i.e.* the corresponding LFSR will have a 64 delay units (the LFSR equation will be: $(X_n = X_{n-16} + X_{n-64})$. Since the length of the data is only 65, generation of any further polynomials by this method will not be of any use. In general, the "squaring" is continued till $2^i d < N$.

From (9) we know that a particular bit of the ciphertext equals the corresponding bit of the LFSR sequence with some probability. The underlying idea of this fast correlation attack is to reconstruct the entire LFSR sequence bit-by-bit, iteratively. For this purpose, a particular bit of the ciphertext is chosen. If, upon examination of all the linear relations involving the bit at this location, it is found that the observed bit satisfies 'most' of them, then it can be reasonably assumed to equal the LFSR bit at that location.

5.2 The underlying statistical model

The number of linear relations that can be generated for a particular bit $X_i, i = i_1$, say, will naturally be restricted by the cipherlength N

available. Each squaring of the polynomial doubles its length and will continue as long as the quantity $2^i d$ is less than N *i.e.* till $i < \lfloor \log_2(\frac{N}{d}) \rfloor$. In other words, the total number of relations obtained

$$T = \sum_{i=0}^{\log_2(N/d)} (N - 2^i d) = N \log_2(\frac{N}{2d}) + d \tag{11}$$

Since every relation is satisfied by all $t+1$ bits, the average number of relations per bit equals

$$m = \frac{(t+1)T}{N} \approx \log_2(\frac{N}{2d})(t+1) \tag{12}$$

Consider the ith bit X_i. These relations may be expressed in the form:

$$L_l = X_i + w_l = 0 \qquad l = 1, \cdots, m, \tag{13}$$

where w_l represents a sum of exactly t different remaining terms with X_i in one of the $t+1$ positions in (2) and also its multiples.

Consider now a bit of the cipherstream, C_i instead of X_i in (13) with

$$L_l = C_i + z_l \qquad l = 1, \cdots, m \tag{14}$$

with z_l representing a sum of exactly t different remaining terms with C_n in one of the $t+1$ positions in (13) and its multiples.

In this case, L_l may not be equal to zero.

Now, let $w_l = w_{l1} + w_{l2} + \cdots + w_{lt}$ and $z_l = z_{l1} + z_{l2} + \cdots + z_{lt}$ where, w_{lj} and z_{lj}, $j = 1, \cdots, t$ are binary variables, all independent and identically distributed with equal probability of being 0 or 1. Note that $P(X_i = C_i) = p = P(w_{lj} = z_{lj})$.

Then, $s(t) = P(w_l = z_l)$ can be recursively computed as follows:

$$\begin{aligned} s(1) &= p \\ s(j) &= ps(j-1) + (1-p)(1-s(j-1)) \qquad j = 2, \cdots, t. \end{aligned} \tag{15}$$

Observe that, for a particular ciphertext bit to satisfy the lth relation i.e. $L_l = C_i + z_l = 0$, either $C_i = X_i$ and $w_l = z_l$ or $C_i \neq X_i$ and $w_l \neq z_l$. Hence

$$\begin{aligned} P(L_1 = \cdots = L_h = 0; L_{h+1} = \cdots = L_m = 1) &= ps^h(1-s)^{m-h} \\ &+ (1-p)(1-s)^h s^{m-h} \end{aligned}$$

where $s = s(t)$. Further

$$\begin{aligned} &P(C_i = X_i | L_1 = \cdots = L_h = 0; = L_{h+1} = \cdots = L_m = 1) \\ &= \frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h} + (1-p)(1-s)^h s^{m-h}} \end{aligned}$$

$$\begin{aligned} &P(C_i \neq X_i | L_1 = \cdots = L_h = 0; = L_{h+1} = \cdots = L_m = 1) \\ &= \frac{(1-p)(1-s)^h s^{m-h}}{ps^h(1-s)^{m-h} + (1-p)(1-s)^h s^{m-h}} \end{aligned}$$

The basic strategy of the attack is as follows. For a bit C_i, we start with an *a priori* probability $p = P(C_i = X_i) > 0.5$. We count the number h of indices l for which $L_l = 0$. We then alter the *a priori* probability $p = P(C_i = X_i)$ to a new value p^* using (16). It is to be expected that if $C_i = X_i$ is true then p^* must increase and vice-versa. This can be verified by computing the expected value of p^* in the two cases.

$$\begin{aligned} &E(p^* | C_i = X_i) \\ &= \sum_{h=0}^{m} \binom{m}{h} \frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h} + (1-p)(1-s)^h s^{m-h}} s^h(1-s)^{m-h} \\ &E(p^* | C_i \neq X_i) \\ &= \sum_{h=0}^{m} \binom{m}{h} \frac{ps^h(1-s)^{m-h}}{ps^h(1-s)^{m-h} + (1-p)(1-s)^h s^{m-h}} s^{m-h}(1-s)^h \qquad (16) \end{aligned}$$

5.3 The Algorithms

Let

$$\begin{aligned} R &= P(c_n = x_n, \quad \text{and } c_n \text{ satisfies at least } h \text{ of } m \text{ relations}), \\ Q &= P(c_n \ \text{ satisfies at least } h \text{ out of } m \text{ relations}), \\ T &= P(c_n = x_n | c_n \ \text{ satisfies at least h of m relations}). \end{aligned}$$

Then, using (16)

$$Q = \sum_{i=h}^{m} \binom{m}{i} (ps^i(1-s)^{m-i} + (1-p)(1-s)^i s^{m-i}) \qquad (17)$$

$$R = \sum_{i=h}^{m} \binom{m}{i} ps^i(1-s)^{m-i}, \qquad T = R/Q \qquad (18)$$

The quantity h/m which is the minimum fraction of equations that a bit of the cipherstream must satisfy, shall be henceforth, referred to as the upper threshold.

Further, let

$$\begin{aligned} V &= P(c_n = x_n, \text{and } c_n \text{ satisfies at most } h \text{ of } m \text{ relations}), \\ W &= P(c_n \neq x_n, \text{and } c_n \text{ satisfies at most } h \text{ of } m \text{ relations}), \\ D &= P(c_n \text{ satisfies at most } h \text{ out of } m \text{ relations}), \\ E &= P(c_n \neq x_n / c_n \text{ satisfies at most h of m relations}) \end{aligned}$$

Then, using (16)

$$D = \sum_{i=0}^{h} \binom{m}{i} (ps^i(1-s)^{m-i} + (1-p)(1-s)^i s^{m-i}) \qquad (19)$$

$$V = \sum_{i=0}^{h} \binom{m}{i} ps^i(1-s)^{m-i}, \qquad (20)$$

$$W = \sum_{i=0}^{h} \binom{m}{i} (1-p)s^i(1-s)^{m-i}, \quad E = W/D \qquad (21)$$

The maximum fraction of equations h/m, that a bit can satisfy in order to be designated as wrong shall be called the lower threshold. Note that the value of h for the upper threshold is different from that of h for the lower threshold.

Meier and Staffelbach [9] give two algorithms based on these computations. One is an exponential-time attack which is non-iterative in nature. It has limited scope as it has been seen that for $t \geq 10$ and $p \leq 0.75$, this algorithm holds no advantage over an exhaustive search of the initial conditions.

The other algorithm is polynomial time. It starts with a value of h such that the relative increase of correct bits, given by $W - V$ is maximum and a threshold $N = UN$ which is the expected number of bits with $p^* < p_{threshold}$. The value of p^* is calculated from which the number of bits with p^* less than a threshold, *i.e.* N_w is counted. If this is greater than $N_{threshold}$, only the bits with p^* less than the threshold are complemented and the procedure continued till all the bits equal those of the cipherstream. However, if N_w is less than the threshold, the algorithm must restart with a new *a priori* probability.

It is seen that the polynomial time algorithm stabilizes in only a few iterations.

5.4 A modification of the Meier-Staffelbach algorithm

We now consider an algorithm that uses the relationships derived in the last section to obtain some bits of the LFSR sequence. Once a sufficient number of bits have been correctly determined (slightly more than the length of the LFSR), the initial conditions of the corresponding LFSR are obtained by constructing and solving a system of linear equations.

Simulations show that as the upper threshold is increased, the probability of correctly determining the bits increases while the number of bits correctly determined decreases. The reverse situation occurs as the lower threshold is increased, the probability that a bit is wrong decreases while the number of wrong bits increases. It can hence be concluded that the thresholds must be chosen with a trade-off in mind, that of ensuring a particular probability of correct determination of the bits and at the same time, obtaining some required number of them.

The algorithm can now be summarized as follows. Using (17) and (18), obtain the upper threshold such that the probability of correct determination is at least 0.95 and the number of bits correctly determined is at least equal to d, the length of the LFSR. Using (21) and (11), obtain the lower threshold such that the probability of wrongful determination is at least 0.95. Complement these bits. Express the bits thus determined in terms of the initial conditions of the LFSR and solve the resultant linear system in order to recover the initial conditions.

Let us consider an example to illustrate the approach. Consider, once again, the polynomial $1 + x + x^4$. Let the LFSR sequence be $X_0, X_1, X_2, X_3, X_4, \cdots$. Note that X_0, X_1, X_2 and X_3 represent the initial conditions of the LFSRs. Suppose that we have been able to determine the bits X_4, X_8, X_{10}, X_{12}. Then, expressing each of them in terms of the initial conditions, we have the following system of equations:

$$\begin{pmatrix} X_4 \\ X_8 \\ X_{10} \\ X_{12} \end{pmatrix} = \begin{pmatrix} 1001 \\ 0111 \\ 0101 \\ 1100 \end{pmatrix} \begin{pmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

Since the left-hand-side vector is completely known, determining the vector on the right-hand-side can be achieved using standard algorithms.

Do this for all combinations of the identified bits, taken d at a time. Note that the system may not always be solvable in which case that particular combination of bits must be rejected. From a plot of the frequency distribution of the occurrence of the initial conditions determined in the above step, locate the set/sets occurring most frequently. When this yields only one set, it can be safely assumed that this is the correct initial condition. If there are multiple sets, the Hamming distance between the LFSR output corresponding to each of these and the cipherstream must be computed. The set with the lowest Hamming distance will be the desired initial condition.

This approach has the chief advantage of being simple yet, produces satisfactory results. It performs well even for smaller cipherlengths though the number of taps must be small.

6 Other fast correlation attacks

After the pioneering work of Meier and Staffelbach, various algorithms for fast correlation attacks have been published. All of them have two basic components: a method of obtaining low-density (small number of taps) parity checks, and secondly, an iterative error-correction algorithm. Described here is one such algorithm, which was proposed in [21] and improved in [22]. During each iteration, parity-checks are calculated bit-by-bit at first. This is followed by employing Bayesian bit-by-bit error correction based on the estimation of the relevant posterior probabilities obtained using posterior probabilities from the previous iteration as the prior probabilities in the current one.

A parity-check is any linear relationship satisfied by an LFSR sequence. Apart from the technique used by Meier and Staffelbach ([9], [25]) in Section 2, techniques using polynomial residues [22], or discrete log computations [10]. A set of polynomial multiples such that no power polynomial appears in more than one polynomial multiple is called a set of orthogonal parity-check polynomials.

Let $\Pi_i = \pi_k(i), i = 1, 2, \ldots, |\Pi_i|$, where $|\Pi_i|$ denotes the cardinality of $\prod_i$, be a set of orthogonal parity-checks related to the ith bit, generated through polynomial multiples of $c(x)$. Let $w = t - 1$ where t is the number of taps. Let a parity-check value be defined as $\alpha_k(i) = \sum_{l \in \pi_k(i)} C_i$. Let

$$p_i = P\left(E_i = 1 | \{C(i)\}_{k=1}^{|\prod_i|} = \{c(i)\}_{k=1}^{|\prod_i|}\right)$$
$$= \frac{q_i \prod_{l=1}^{|\prod_i|} q_l(i)^{\overline{c_l}(i)}(1 - q_l(i))^{c_l(i)}}{q_i \prod_{l=1}^{|\prod_i|} q_l(i)^{\overline{c_l}(i)}(1 - q_l(i))^{c_l(i)} + (1 - q_i) \prod_{i=1}^{|\prod_i|}(1 - q_l(i))^{\overline{c_l}(i)} q_l(i))^{c_l(i)}} \tag{22}$$

where p_i and q_i are the posterior and prior probabilities for the current iteration, $\overline{c_l}(i) = 1 - c_l(i)$, $q_l(i) = (1 - \prod_{t=1}^{w}(1 - 2q_{m_t}))/2$ and $\{m_t\}_{t=1}^{w}$ denotes the set of indices of the bits involved in the parity-check $\pi_l(i)$, for any $l = 1, 2, \ldots, |\prod_i|$ and $i = 1, 2, \ldots, N$.

The algorithm starts by calculation of the parity-checks of each bit of the cipher stream. The posterior probabilities p_i are calculated using (22). The algorithm complements the bit if p_i exceeds 0.5. The posterior probabilities of the current iteration are used as the prior probabilities of the next one. If the empirical error rate exceeds a preset tolerance, the algorithm must be restarted with fresh parity-checks. The algorithm terminates when all parity-checks are satisfied.

A number of fast correlation attacks can be found in [10, 11, 12, 13, 15, 16, 17, 18, 19, 20].

7 The Decimation Attack

This approach [23] of determining the LFSR initial conditions through a decimated version of the ciphertext can be very useful provided the appropriate decimated lengths can be obtained.

Consider a decimation of the LFSR output sequence X_i, $i = 1, 2, \cdots, N$ by a factor of r resulting in the sequence X_D, *i.e.* $X_D =$

$X[r]$. Then the ***simulated LFSR*** or the LFSR producing X_D, has the properties:

1. The feedback polynomial C^* of the simulated LFSR is the minimumm polynomial of β^r in $GF(q^d)$.
2. The period T_p^* of $C^*(x)$ equals $\frac{T_p}{gcd(r,T_p)}$.
3. The degree d^* of $C^*(x)$ equals the multiplicative order of q in Z_T.

Further, all $d \in F_k$, where $F_k = k, kq, kq^2, \cdots modT_P$ stands for the cyclotomic coset of $kmoduloT_p$, give rise to the same simulated LFSR, though with different initial conditions. Every sequence produced by the simulated LFSR equals X_D for some set of initial conditions of the original LFSR.

Let $C_i[r]$ be the decimated ciphertext. Then the simulated LFSR producing this can be determined using Siegenthaler's approach, requiring $2^{d^*} - 1$ searches. Now, a d^*-bit candidate can be used to generate d bits of the decimated sequence. Each bit of this sequence also satisfies the undecimated sequence X. Hence a system of d equations can be constructed involving the known d^* bits. This system will have rank $d - d^*$ which can be further expressed in terms of a different set of d^* bits. An exhaustive search over these d^* parameters, using the ciphertext and Siegenthaler's approach, determines the correct d length initial conditions of the actual LFSR.

8 Function estimation for nonlinear combiners with memory

In order to overcome the trade-off between the linear complexity and correlation immunity of combining functions [24], the use of combiners with memory was suggested [5] and analysed extensively in [25, 26]. This makes decryption extremely difficult particularly when the combining function is unknown.

Let the output of the combiner be

$$Y_i = f(\boldsymbol{X}_i, Y_{i-1}), \qquad i = 2, 3, \ldots, N. \tag{23}$$

where $\boldsymbol{X}_i = (X_i^1, X_i^2, \ldots, X_i^m)$, is the vector of outputs of the m LFSRs. It is assumed that the LFSR polynomials as well as the initial conditions are known, but the binary combining function f is unknown. Since the LFSR outputs are completely known, we exclude it from the notations of the following analysis which is conditional on these. The task of determining the combining function amounts to choosing the right truth table of the function f from among the possible 2^{m+1} truth tables.

The likelihood of the observed cipher-text is

$$P(C_N, C_{N-1}, \ldots, C_1) = P(C_1) \prod_{i=2}^{N} P(C_i | C_{i-1}, \ldots, C_1).$$

When the combining function is unknown but memoryless, we approximated $P(C_i|C_{i-1}, \ldots, C_1)$ by $P(C_i)$ (see Section 4). This amounts to ignoring the weak dependence of the ciphertext bits which may result from the dependence of the plaintext bits. In the present case however, successive bits may be dependent because of the memory of the combiner. Therefore, we use the more realistic approximation $P(C_i|C_{i-1}, \ldots, C_1) = P(C_i|C_{i-1})$, that is,

$$P(C_N, C_{N-1}, \ldots, C_1) = P(C_1) \prod_{i=2}^{N} P(C_i|C_{i-1}). \tag{24}$$

A justification of this approximation is given by Palit and Dasgupta [27], who show that the ciphertext is a first order Markov process if the coded plaintext bits are independent.

It can be seen that

$$P(Y_{i-1}|C_{i-1}) \quad = \quad \begin{cases} (1-p_0)C_{i-1} + p_0(1 - C_{i-1}) & \text{if } Y_{i-1} = 0, \\ p_0 C_{i-1} + (1-p_0)(1 - C_{i-1}) & \text{if } Y_{i-1} = 1, \end{cases} \tag{25}$$

$$\begin{aligned} P(C_i|Y_{i-1}) &= [1-p_0+(2p_0-1)f(\boldsymbol{X}_i,Y_{i-1})]^{C_i} \\ &\qquad \cdot [p_0-(2p_0-1)f(\boldsymbol{X}_i,Y_{i-1})]^{1-C_i}. \end{aligned} \tag{26}$$

Substituting in the expression $P(C_i|C_{i-1}) = \sum_{Y_{i-1}} P(C_i|Y_{i-1})P(Y_{i-1}|C_{i-1})$, and plugging into (2), we have the approximate likelihood

$$\begin{aligned} &\prod_{i=2}^{N} P(C_i|C_{i-1}) \\ &= \prod_{j=0}^{2^m-1} \prod_{i:\boldsymbol{X}_i = \boldsymbol{x}^j} \{[1-p+(2p-1)f(\boldsymbol{x}^j,0)]^{C_i} \\ &\quad \cdot [p-(2p-1)f(\boldsymbol{x}^j,0)]^{1-C_i} \quad \cdot \quad [(1-p_0)C_{i-1}+p_0(1-C_{i-1})] \\ &\quad + [1-p_0+(2p_0-1)f(\boldsymbol{x}^j,1)]^{C_i} \cdot [p_0-(2p_0-1)f(\boldsymbol{x}^j,1)]^{1-C_i} \\ &\qquad \cdot [p_0C_{i-1}+(1-p_0)(1-C_{i-1})]\}, \end{aligned} \tag{27}$$

where $\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{2^m-1}$ are the 2^m possible values of the $\boldsymbol{X}_i$s. The advantage of this approximate likelihood is that the factor corresponding to each $\boldsymbol{x}^j$ can be maximized separately. This factor has to be maximized simultaneously with respect to binary parameters $f(\boldsymbol{x}^j, 0)$ and $f(\boldsymbol{x}^j, 1)$. Let ℓ_{jkl} be the value of this factor for $f(\boldsymbol{x}^j, 0) = k$ and $f(\boldsymbol{x}^j, 1) = l$, $k, l = 0, 1$. Then we have

$$\begin{aligned} \ell_{j00} &= \prod_{i:\boldsymbol{X}_i = \boldsymbol{x}^j} \{(1-p)^{C_i} \cdot p^{1-C_i} \cdot [(1-p)C_{i-1}+p(1-C_{i-1})] \\ &\qquad + (1-p)^{C_i} \cdot p^{1-C_i} \cdot [pC_{i-1}+(1-p)(1-C_{i-1})]\} \\ &= (1-p)^{N_{j10}+N_{j11}} \cdot p^{N_{j00}+N_{j01}}, \end{aligned} \tag{28}$$

where, N_{jkl} is the number of bits (i) for which $\boldsymbol{X}_i = \boldsymbol{x}^j$, $C_i = k$ and $C_{i-1} = l$, $k, l = 0, 1$. Similarly we have the simplifications

$$\ell_{j01} = [p^2+(1-p)^2]^{N_{j00}+N_{j11}} \cdot [2p(1-p)]^{N_{j01}+N_{j10}}, \tag{29}$$

$$\ell_{j10} = [2p(1-p)]^{N_{j00}+N_{j11}} \cdot [p^2+(1-p)^2]^{N_{j01}+N_{j10}}, \tag{30}$$

$$\ell_{j11} = p^{N_{j10}+N_{j11}} \cdot (1-p)^{N_{j00}+N_{j01}}. \tag{31}$$

If $\ell_{jk_*l_*}$ is the largest of the ℓ_{jkl}s for $k, l = 0, 1$, then the approximate maximum likelihood estimator of $f(x^j, 0)$ and $f(x^j, 1)$ are k_* and l_*, respectively.

A theoretical performance analysis can be don along the following lines. Note that logarithms of the ℓ_{jkl}s are linear functions of N_{j00}, N_{j01}, N_{j10} and N_{j11}. We shall argue that the joint distribution of these variables is approximately normal. Let N_{jn} be the number of bits (i) such that $X_i = x^j$ and $Y_{i-1} = n$. It is clear that

$$N_{jn} = N_{j00n} + N_{j01n} + N_{j10n} + N_{j11n},$$

where N_{jkln} is the number of bits (i) such that $X_i = x^j$, $C_i = k$, $C_{i-1} = l$ and $Y_{i-1} = n$. Since Y_{i-1} is not observable, none of the counts mentioned in the above equation is observable. However, the counts

$$N_{jkl} = N_{jkl0} + N_{jkl1}, \quad k = 0, 1, \ \ l = 0, 1, \ \ j = 0, 1, \ldots, 2^m - 1,$$

are observable.

Given N_{j0}, the allocation to its constituent parts follows a distribution which is approximately multinomial, and can be further approximated by a normal distribution with matching mean vector and covariance matrix. The same can be said about the constituents of N_{j1}, which are conditionally independent of the components of N_{j0}. Thus, the conditional distribution of N_{j00}, N_{j01}, N_{j10} and N_{j11} given N_{j0} and N_{j1} is approximately multivariate normal. This should in principle provide a way of obtaining approximate expressions for the probability of correct identification of $f(x^j, 0)$ and $f(x^j, 1)$. However, the expressions will depend on the true values of these binary parameters. Therefore, we need to consider four special cases corresponding to the possible values of this pair of parameters.

Case I: $f(x^j, 0) = 0$, $f(x^j, 1) = 0$.

Correct identification in this case corresponds to the event

$$\begin{aligned} a_{00} &= \log \ell_{j00} - \log \ell_{j11} > 0, && (32)\\ b_{00} &= \log \ell_{j00} - \log \ell_{j10} > 0, && (33)\\ c_{00} &= \log \ell_{j00} - \log \ell_{j01} > 0. && (34)\end{aligned}$$

Note that a_{00}, b_{00} and c_{00} are linear functions of N_{j00}, N_{j01}, N_{j10} and N_{j11}. Specifically, $(a_{00} : b_{00} : c_{00})^T = Q_{00}(N_{j00} : N_{j01} : N_{j10} : N_{j11})^T$, where

$$Q_{00} = \begin{pmatrix} \log\frac{p}{2pq} & \log\frac{p}{p^2+q^2} & \log\frac{q}{p^2+q^2} & \log\frac{q}{2pq} \\ \log\frac{p}{p^2+q^2} & \log\frac{p}{2pq} & \log\frac{q}{2pq} & \log\frac{q}{p^2+q^2} \\ \log\frac{p}{q} & \log\frac{p}{q} & \log\frac{q}{p} & \log\frac{q}{p} \end{pmatrix}, \qquad (35)$$

where $q = 1 - p$. In this case, we have for given N_{j0},

$$E\left(\begin{pmatrix} N_{j000} \\ N_{j010} \\ N_{j100} \\ N_{j110} \end{pmatrix} | N_{j0}\right) = N_{j0}\begin{pmatrix} p^2 \\ pq \\ pq \\ q^2 \end{pmatrix}.$$

On the other hand,

$$\begin{aligned} P(C_i = 0, C_{i-1} = 0 | Y_{i-1} = 0) &= p^2, \\ P(C_i = 0, C_{i-1} = 1 | Y_{i-1} = 0) &= pq, \\ P(C_i = 1, C_{i-1} = 0 | Y_{i-1} = 0) &= pq, \\ P(C_i = 1, C_{i-1} = 1 | Y_{i-1} = 0) &= q^2, \end{aligned}$$

so that the conditional variance-covariance matrix is

$$Cov\left(\begin{pmatrix} N_{j000} \\ N_{j010} \\ N_{j100} \\ N_{j110} \end{pmatrix} | N_{j0}\right) = N_{j0}\begin{pmatrix} p^2 & 0 & 0 & 0 \\ 0 & pq & 0 & 0 \\ 0 & 0 & pq & 0 \\ 0 & 0 & 0 & q^2 \end{pmatrix} - N_{j0}\begin{pmatrix} p^2 \\ pq \\ pq \\ q^2 \end{pmatrix}\begin{pmatrix} p^2 \\ pq \\ pq \\ q^2 \end{pmatrix}^T.$$

Likewise, for given N_{j1}, we have

$$E\left(\begin{pmatrix} N_{j001} \\ N_{j011} \\ N_{j101} \\ N_{j111} \end{pmatrix} | N_{j1}\right) = N_{j1}\begin{pmatrix} pq \\ p^2 \\ q^2 \\ pq \end{pmatrix}.$$

and

$$Cov\left(\begin{pmatrix} N_{j001} \\ N_{j011} \\ N_{j101} \\ N_{j111} \end{pmatrix} |N_{j1}\right) = N_{j1}\begin{pmatrix} pq & 0 & 0 & 0 \\ 0 & p^2 & 0 & 0 \\ 0 & 0 & q^2 & 0 \\ 0 & 0 & 0 & pq \end{pmatrix} - N_{j1}\begin{pmatrix} pq \\ p^2 \\ q^2 \\ pq \end{pmatrix}\begin{pmatrix} pq \\ p^2 \\ q^2 \\ pq \end{pmatrix}^T$$

In order to compute the parameters of the normal approximation of the joint distribution of N_{j00}, N_{j01}, N_{j10} and N_{j11} conditional on N_{j0} and N_{j1}, we need to combine the above results. The mean vector is

$$E\left(\begin{pmatrix} N_{j00} \\ N_{j01} \\ N_{j10} \\ N_{j11} \end{pmatrix} |N_{j0}, N_{j1}\right) = \begin{pmatrix} N_{j0}p^2 + N_{j1}pq \\ N_{j0}pq + N_{j1}p^2 \\ N_{j0}pq + N_{j1}q^2 \\ N_{j0}q^2 + N_{j1}pq \end{pmatrix}.$$

The corresponding covariance matrix is

$$\begin{pmatrix}
\begin{array}{c} N_{j0}p^2 + N_{j1}pq \\ -N_{j0}p^4 - N_{j1}p^2q^2 \end{array} & \begin{array}{c} -N_{j0}p^3q \\ -N_{j1}p^3q \end{array} & \begin{array}{c} -N_{j0}p^3q \\ -N_{j1}pq^3 \end{array} & \begin{array}{c} -N_{j0}p^2q^2 \\ -N_{j1}p^2q^2 \end{array} \\
\begin{array}{c} -N_{j0}p^3q \\ -N_{j1}p^3q \end{array} & \begin{array}{c} N_{j0}pq + N_{j1}p^2 \\ -N_{j0}p^2q^2 - N_{j1}p^4 \end{array} & \begin{array}{c} -N_{j0}p^2q^2 \\ -N_{j1}p^2q^2 \end{array} & \begin{array}{c} -N_{j0}pq^3 \\ -N_{j1}p^3q \end{array} \\
\begin{array}{c} -N_{j0}p^3q \\ -N_{j1}pq^3 \end{array} & \begin{array}{c} -N_{j0}p^2q^2 \\ -N_{j1}p^2q^2 \end{array} & \begin{array}{c} N_{j0}pq + N_{j1}q^2 \\ -N_{j0}p^2q^2 - N_{j1}q^4 \end{array} & \begin{array}{c} -N_{j0}pq^3 \\ -N_{j1}pq^3 \end{array} \\
\begin{array}{c} -N_{j0}p^2q^2 \\ -N_{j1}p^2q^2 \end{array} & \begin{array}{c} -N_{j0}pq^3 \\ -N_{j1}p^3q \end{array} & \begin{array}{c} -N_{j0}pq^3 \\ -N_{j1}pq^3 \end{array} & \begin{array}{c} N_{j0}q^2 + N_{j1}pq \\ -N_{j0}q^4 - N_{j1}p^2q^2 \end{array}
\end{pmatrix}.$$

The probability of correct identification of $f(x^j, 0)$ and $f(x^j, 1)$ is the joint probability of the events (32 - 34), which may be computed from (35) and the distribution of $(N_{j00} : N_{j01} : N_{j10} : N_{j11})^T$ described above.

In order to get a better understanding of the probability of correct estimation, let us consider the special case $N_{jn} = 2^{-m-1}N$ for $n = 0, 1, \ldots, 2^m$ and $n = 0, 1$. In this case, the covariance matrix of $(N_{j00} : N_{j01} : N_{j10} : N_{j11})^T$ simplifies further and can be factored as

$2^{-m-1}N\boldsymbol{B}\boldsymbol{B}^T$, where

$$\boldsymbol{B} = (pq)^{1/2}\begin{pmatrix} 1 & 0 & 2^{1/2}p \\ 0 & 1 & -2^{1/2}p \\ -1 & 0 & 2^{1/2}q \\ 0 & -1 & -2^{1/2}q \end{pmatrix}.$$

Consequently

$$\begin{aligned} E\begin{pmatrix} a_{00} \\ b_{00} \\ c_{00} \end{pmatrix} &= \frac{N}{2^{m+1}}\boldsymbol{Q}_{00}\begin{pmatrix} p \\ p \\ q \\ q \end{pmatrix} \\ &= \frac{N}{2^{m+1}}\begin{pmatrix} 2p\log p + 2q\log q - \log(2pq(p^2+q^2)) \\ 2p\log p + 2q\log q - \log(2pq(p^2+q^2)) \\ 2(p-q)\log\frac{p}{q} \end{pmatrix}, \\ Cov\begin{pmatrix} a_{00} \\ b_{00} \\ c_{00} \end{pmatrix} &= 2^{-m-1}N(\boldsymbol{Q}_{00}\boldsymbol{B})(\boldsymbol{Q}_{00}\boldsymbol{B})^T, \end{aligned}$$

and $\boldsymbol{Q}_{00}\boldsymbol{B}$ simplifies to

$$\boldsymbol{Q}_{00}\boldsymbol{B} = (pq)^{1/2}\begin{pmatrix} \log\frac{p^2+q^2}{2q^2} & -\log\frac{p^2+q^2}{2p^2} & 2^{1/2}(p-q)\log\frac{p^2+q^2}{2pq} \\ -\log\frac{p^2+q^2}{2p^2} & \log\frac{p^2+q^2}{2q^2} & -2^{1/2}(p-q)\log\frac{p^2+q^2}{2pq} \\ \log\frac{p^2}{q^2} & -\log\frac{p^2}{q^2} & 0 \end{pmatrix}.$$

Consider the further special case where $p = 0.5 + \delta$ where δ is a small number compared to 1. Then we have the approximations

$$E\begin{pmatrix} a_{00} \\ b_{00} \\ c_{00} \end{pmatrix} \approx \frac{N}{2^{m+1}}\begin{pmatrix} 4\delta^2 \\ 4\delta^2 \\ 8\delta^2 \end{pmatrix}, \quad Cov\begin{pmatrix} a_{00} \\ b_{00} \\ c_{00} \end{pmatrix} \approx \frac{N}{2^{m+1}}\begin{pmatrix} 8\delta^2 & 8\delta^3 & 16\delta^2 \\ 8\delta^3 & 8\delta^2 & 16\delta^2 \\ 16\delta^2 & 16\delta^3 & 32\delta^2 \end{pmatrix}$$

It can be seen that a_{00} and b_{00} are almost uncorrelated and $c_{00} \approx 2a_{00} \approx 2b_{00}$. Hence, the probability of correct estimation of $f(x^j, 0)$ and $f(x^j, 1)$ is approximately

$$P(a_{00} > 0, b_{00} > 0, c_{00} > 0) = P(a_{00} > 0) = \Phi(\delta 2^{-m/2}N^{1/2}), \qquad (36)$$

The analyses for the other three cases are done in a similar fashion, by Palit and Dasgupta [27] using a series of approximations and supporting simulations. The results show that the cipherlength required to achieve a given probability of correct estimation of the pair $f(x^j, 0)$ and $f(x^j, 1)$ is of the order of $|p_0 - 1/2|N^{1/2}$ when $f(x^j, 0) = f(x^j, 1)$ and of the order of $|p_0 - 1/2|^2 N^{1/2}$ when $f(x^j, 0) \neq f(x^j, 1)$. The latter expression clearly explains why a very large cipherlength may be needed when the unknown combiner has memory. Also provided in [27] is an alternative approach based on the fact that $P(M_i = M_{i-1})$ is generally different from one-half for most coded plaintext messages. This algorithm seems to perform as well as the one described above.

9 Conclusion

Cryptanalysis of stream cipher model where the combining function has memory of more than one bit seems quite complicated and cumbersome. An elegant statistical approach is looked for from statisticians. Also different statistical methods may be adopted to reduce the computational complexity of the cryptanalysis. Besides the stream cipher model described in this paper, there are plenty of other models. Some of them have beeen or are being cryptanalyzed; as the basic tool is statistics for such analysis. This area demands the attention from the statistics community.

References

[1] **C.E. Shannon**, (1949). Communication Theory of secrecy systems, *Bell Systems Technical Journal*, **28**, 656-715.

[2] **A. Menezes, P. van Oorschot and S. Vanstone**, (1997). *Handbook of applied cryptography*, CRC Press.

[3] **J.L. Massey**, (1969). Shift register synthesis and BCH decoding," *IEEE Trans. on Information Theory*, **IT-15**, 122-127.

[4] **E.R. Berlekamp**, (1968). *Algebraic Coding Theory*,New York: McGraw-Hill, ch. 7,10.

[5] **R.A. Rueppel**, (1986). *Analysis and Design of Stream ciphers*, Springer Verlag.

[6] **T. Siegenthaler**, (1985). Decrypting a class of stream ciphers using ciphertext only, *IEEE Transactions on Computers*, **c-34**, No.1, 81-85.

[7] **S. Palit and B. K. Roy**, (1999). Cryptanalysis of LFSR-Encrypted Codes with unknown combining function, *Advances in Cryptology-ASIACRYPT '99*, Lecture Notes in Computer Science, 1716, K.Y. Lam. E. Okamoto, C. Xing Eds., Springer-Verlag, 306-320.

[8] **B.K. Roy**, (1998). Ciphertext only cryptanalysis of LFSR based encryption schemes, *Proceedings of the National Seminar on Cryptology*, Delhi, A-19–A-24.

[9] **W. Meier and O. Staffelbach**, (1989). Fast correlation attacks on certain stream ciphers, *Journal of Cryptology*, **1**, no.3, 159 - 176.

[10] **K. Zeng and M. Huang**, (1990). On the linear syndrome method in cryptanalysis, *Advances in Cryptology - CRYPTO '88*, **403**, S. Goldwasser, editor, Springer Verlag, 469–478.

[11] **V. Chepyzhov and B. Smeets**, (1991). On a fast correlation attack on stream ciphers, *Advances in Cryptology - EUROCRYPT '91*, **547**, D.W. Davies, editor,

Springer Verlag, 176-185.

[12] **A. Clark, J. Golic and E. Dawson,** (1996). A comparison of fast correlation attacks, *Fast Software Encryption,* Third International Workshop (LNCS 1039), D. Gollman, editor, Springer Verlag, 145-157.

[13] **R. Forre,** (1990). A fast correlation attack on nonlinearly feedforward filtered shift register sequences, *Advances in Cryptology - EUROCRYPT '89* (LNCS 434), 586-95.

[14] **M.J. Mihaljevic and J. Golic,** (1990). A fast iterative algorithm for a shift register initial state reconstruction given the noisy output sequence, *Advances in Cryptology - AUSCRYPT '90,* **453**, J. Seberry and J. Pieprzyk, editors, Springer Verlag, 165-175.

[15] **M.J. Mihaljevic and J. Golic,** (1991). A comparison of cryptanalytic principles based on iterative error-correction, *Advances in Cryptology - EUROCRYPT '91,* **547**, D.W. Davies, editor, Springer Verlag, 527-531.

[16] **T. Johansson and F. Jönsson,** (1999). Improved fast correlation attacks on stream ciphers via convolutional codes, *Proceedings of Cryptology - EUROCRYPT '99,* Springer Verlag, LNCS 1592, 347-362.

[17] **T. Johansson and F. Jönsson,** (1999). Fast correlation attacks based on Turbo Code techniques," *Proceedings of Cryptology - Crypto '99,* Springer Verlag, LNCS 1666, 181-197.

[18] **V.V. Chepyzhov, T. Johansson and B. Smeets,** (2000). A simple algorithm for fast correlation attacks on stream ciphers, *Fast Software Encryption.*

[19] **A. Canteaut and M. Trabbia**, (2000). Improved Fast Correlation Attacks Using Parity-Check Equations of Weight 4 and 5, *EUROCRYPT 2000* , LNCS 1807, B. Preneel Ed., Springer-Verloag, Berlin Heidelberg.

[20] **M. Mihaljević, M. P. C. Fossorier and H. Imai**, (2000). Fast Correlation Attack Algorithm with List Decoding and an Application," *Fast Software Encryption-FSE 2000.*

[21] **M.J. Mihaljević and J. Dj. Golić**, (1991). A comparison of cryptanalytic principles based on iterative error correction, *Advances in Cryptology-EUROCRYPT '91*, Lecture Notes in Computer Science, vol.547, D.W. Davies, ed., Springer Verlag, 527-531.

[22] **J. Dj. Golić, M. Salmasizadeh, A. Clark, A. Khodkar and E. Dawson**, (1996). Discrete Optimisation and Fast Correlation Attacks," *Cryptographic Policy and Algorithms-Brisbane '95*, Lecture Notes in Computer Science, E. Dawson and J. Golić eds., Springer-Verlag, 188-202.

[23] **E. Filiol**, (2000). Decimation Attack of Stream Ciphers," *Progress in Cryptology - INDOCRYPT 2000*, Lecture Notes in Computer Science, 1977, B. Roy, E. Okamoto Eds., Springer-Verlag, 31-42.

[24] **T. Siegenthaler**, (1984). Correlation-Immunity of Nonlinear Combining functions for Cryptographic Applications, *IEEE Transactions on Information Theory*, **30**, No. 5, 776 - 780.

[25] **W. Meier and O. Staffelbach**, (1992). Correlation properties of combiners with memory in stream ciphers,

Journal of Cryptology, **5**, 67-86.

[26] **J. Dj. Golić**, (1996). Correlation propertiesof a general binary combiner with memory, *Journal of Cryptology*, **9(2)**, 111-126.

[27] **S. Palit and A. Dasgupta**, Determination of combining functions with memory in LFSR based stream ciphers, *Technical Report No. CVPR/03/01*, Indian Statistical Institute, Kolkata. (unpublished).

Bimal Roy
Indian Statistical Institute,
203, B. T. Road, Kolkata 700 108.
email{bimal, sarbanip}@isical.ac.in

Sarbani Palit,
Indian Statistical Institute,
203, B. T. Road, Kolkata 700 108.
email{bimal, sarbanip}@isical.ac.in