Cryptanalysis of LFSR-Encrypted Codes with Unknown Combining Function

Sarbani Palit[1] and Bimal K. Roy[2]

[1] Computer Vision & Pattern Recognition Unit, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, INDIA
sarbanip@isical.ac.in
[2] Applied Statistics Unit, Indian Statistical Institute,
203, B T Road, Calcutta 700 035, INDIA
bimal@isical.ac.in

Abstract. This paper proposes an approach for the cryptanalysis of stream ciphers where the encryption is performed by multiple linear feedback shift registers (LFSR) combined by a nonlinear function. The attack assumes no knowledge of either the LFSR initial conditions or the combining function. Thus, the actual architecture of the encryption system can be arbitrary. The attack is also generalized for the situation when the combining function is correlation immune of any particular order. This is in direct contrast with the existing methods which depend heavily not only on the correlation between the output of a particular LFSR and the ciphertext but also on the actual configuration of the encryption system used. Thus, the proposed method is the first *ciphertext only* attack in the true sense of the phrase. The paper also gives theoretical estimates of the cipherlengths involved in the determination of the initial conditions as well as estimation of the combining function.

1 Introduction

Stream ciphers form an important class of encryption algorithms. Linear feedback shift registers (LFSR) have commonly been used in the keystream generators of such ciphers for a reasons such as suitability for hardware implementation and desirable statistical properties [1]. A popular form of the running key generator is constructed by applying a nonlinear combining function to the outputs of several LFSRs of various lengths. In a general situation, the decrypter may be faced with one or more of the following problems, *viz.* unknown initial conditions of the LFSRs, unknown shift register lengths and polynomials, unknown combining function, availability of limited cipher length, and need for computation in reasonable time,

The problem of determining the unknown LFSR sequences or their initial conditions from available ciphertext using a correlation attack was first explored by Siegenthaler in [2]. This approach exploited the inherent weakness of keystreams generated using LFSRs where knowledge about the LFSR creeps into the encrypted data. It was followed by other forms of correlation attacks

K. Y. Lam, E. Okamoto and C. Xing (Eds.): ASIACRYPT'99, LNCS 1716, pp. 306–320, 1999.

as well as modifications for improvement in speed, see [3,4,5,6,7]. All these approaches, however, implicitly assume that the ciphertext is directly correlated to the LFSR sequences, *i.e.* the combining function is not correlation immune [8] and also *known* to the decrypter.

This paper presents a decryption strategy based on a modification of Siegenthaler's method. The method given here is developed in steps in sections 2 to 5 to include the cases of unknown LFSR initial conditions and unknown combining function, which is eventually allowed to be correlation immune of a particular order. The method is therefore, independent of the actual architecture of the encryption system, unlike the other existing methods in the literature. Though knowledge of the LFSR polynomials is assumed, this assumption can be dispensed with as explained later. The paper also presents a framework for determining the cipherlength requirements involved.

2 Determination of the Initial Conditions

Let N denote the cipherlength available, C the ciphertext, X_i the sequence produced by the ith LFSR, m the total number of LFSRs and d_i, the size of the ith LFSR. Also, $P(A)$ denotes the probability of the occurrence of the event A. Let M be the coded message text, Y be the output of the combining function. We assume initially that the shift register sizes, the LFSR polynomials and the form of the nonlinear combining function are known.

2.1 Siegenthaler's Approach

Siegenthaler's approach [2] for determining the initial conditions of the LFSRs is based on statistical hypothesis testing. Consider the random sequence

$$Z_i = \begin{cases} 1 & \text{if } C = X_i \\ 0 & \text{if } C \neq X_i \end{cases}$$

It can thus be inferred that $\sum Z_i \sim Bin(N, p)$ where $p = P(C = X_i)$. From the system configuration

$$P(C = X_i) = P(Y = X_i)P(M = 0) + P(Y \neq X_i)P(M \neq 0)$$

If either $P(Y = X_i) = 1/2$ or $P(M = 0) = 1/2$, then $P(C = X_i) = 1/2$. However, for all practical coding schemes, $P(M = 0) \neq 1/2$. For instance, for the popularly used Murray code $P(M = 0) = 0.58$. Also, for a function which is not correlation immune, there is at least one input i for which $P(Y = X_i) \neq 1/2$. In order to break the code, the LFSRs are run with various initial conditions. When the correct initial condition is used to generate the LFSR sequence X_i, $p = P(C = X_i) \neq 1/2$. In contrast, for a wrong initial condition, the sequence X_i is random and uncorrelated with C which implies that $p = 1/2$. Thus, given

an input sequence X_i and C, we can determine whether the corresponding initial condition is correct by testing the following hypotheses:

$$\mathcal{H}_0 : p = 0.5,$$
$$\mathcal{H}_1 : p \neq 0.5.$$

The statistic used for testing $\mathcal{H}_0$ against the alternative $\mathcal{H}_1$ is $\sum Z_i$. Let us assume that the fraction of coincidences between the cipher stream and an input *i.e.* $\frac{\sum Z_i}{N}$ is normally distributed. This assumption is justified by the Central Limit Theorem. According to the usual Neyman-Pearson set-up for hypothesis testing, the decision threshold for $\sum Z_i$ is chosen so that the probability of wrongly rejecting $\mathcal{H}_0$ is restricted to a specified value. However, Siegenthaler proposes that the threshold be chosen so that the probability of wrongly *accepting* $\mathcal{H}_0$ (probability of 'miss', p_m or 1-'power' in the terminology of hypothesis testing) is restricted to a specified value. The probability of the other type of error (probability of 'false alarm', p_f) is not controlled. It would depend on the nature of the combining function and the cipherlength.

2.2 Notion of Success and Failure

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. In this paper, we choose $k = 0$. The same choice was implicitly made by Roy [9]. Of course, all the calculations can be generalized for $k > 0$. Our choice of k implies that 'success' is defined as the case when the shortlist of selected initial conditions contains only the correct one.

2.3 Choice of Optimal Threshold to Maximize Chance of Success

Siegenthaler [2] suggested that the probability of miss (p_m) should be predetermined. In practice however, the choice of p_m must be a compromise to ensure that not too many wrong candidate solutions are selected. According to the notion of success described above, *no* false alarm is acceptable. Thus, instead of using a predetermined p_m, we can determine the threshold in such a way that both p_m and the probability of any false alarm is minimized. If the ith LFSR has size d_i, the probability of no false alarm is $(1 - p_f)^{2^{d_i}-2}$, where p_f is the probability of

false alarm for a *particular* candidate solution. Note that if the decision threshold is moved in such a way that $1-(1-p_f)^{2^{d_i}-2}$ (the probability of having at least one wrong initial condition) is reduced, p_m would increase as a consequence. Therefore, the 'minimax' strategy of minimizing $\max\{p_m,\ 1-(1-p_f)^{2^{d_i}-2}\}$ leads to the solution $1-(1-p_f)^{2^{d_i}-2}=p_m$.

Suppose that δ_i is the 'separation' $P(C=X_i)-0.5$. If $\delta_i>0$, the decision rule is to classify an initial condition as 'correct' when $\frac{1}{N}\sum_{i=1}^{N} Z_i > t$, where t is the decision threshold. The minimax solution described above is equivalent to the following equation for t:

$$1-\Phi\left(\frac{t-0.5}{\sqrt{0.25/N}}\right)^{2^{d_i}-2} = \Phi\left(\frac{t-(0.5+\delta_i)}{\sqrt{((0.5+\delta_i)(0.5-\delta_i))/N}}\right),$$

where $\Phi(\cdot)$ is the standard normal distribution function.

A suitable modification of the above condition can be made when $\delta_i<0$.

2.4 A Modification of Siegenthaler's Method

Let us assume once again that $\delta_i>0$. Let f_{ic} be the observed fraction of coincidences, $\sum Z_i/N$, when the chosen initial condition is correct. Suppose that f_{iw} be 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 in section 2.2) if $f_{iw} \le t < f_{ic}$. However, f_{ic} can be larger than f_{iw} even if both of these are on the same side of t. 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 $\delta_i<0$, the minimizer of f_i over all possible initial conditions should be identified as the correct initial condition.

Henceforth we will refer to Siegenthaler's method with threshold chosen as in Section 2.3 as **Method 1**, and the 'best-is-correct' approach described in this section as **Method 2**.

2.5 Performance of the two Methods

Consider Method 1 with the threshold set at t, and assume $\delta_i>0$ without loss of generality.

$$\begin{aligned} &P(i\text{th input is correctly determined})\\ &\quad = P(\text{the 'correct' LFSR sequences has } f_i > t\\ &\qquad\qquad \text{AND all 'wrong' LFSR sequences have } f_i \le t), \end{aligned}$$

where a 'correct' LFSR sequence corresponds to that generated using the correct initial condition (i.c.). Therefore,
$P(i\text{th input is correctly determined}) = (1-p_m)(1-p_f)^{n_i-1}$, where

$1 - p_m = P(\text{a 'correct' LFSR sequence has } f_i > t) \approx \Phi\left(\frac{.5+\delta_i-t}{\sqrt{0.25/N}}\right)$,

$1 - p_f = P(\text{a 'wrong' LFSR sequence has } f_i \leq t) \approx \Phi\left(\frac{t-.5}{\sqrt{0.25/N}}\right)$,

n_i = total no. of LFSR sequences for input $i = 2^{d_i} - 1$.

Note that the probabilities $(1 - p_m)$ and $(1 - p_f)^{n_i-1}$ are the same because of the minimax strategy of section 2.3. Denoting m as the number of inputs,

$$P(\text{all inputs are correctly determined}) = \prod_{i=1}^{m} \Phi\left(\frac{.5 + \delta_i - t}{\sqrt{0.25/N}}\right)^2, \qquad (1)$$

Next consider Method 2. Let

$$p_i = P(C = X_i | i\text{th LFSR i.c. is chosen correctly}).$$

, which can be computed for a particular combining function. The number of coincidences N_{ic} when the i.c. is chosen correctly has a binomial distribution: $N_{ic} \sim Bin(N, p_i)$. When the i.c. is chosen wrongly, the number of coincidences, $N_i \sim Bin(N, 0.5)$. There are $2^{d_i} - 2$ wrong initial conditions. Assuming that $\delta_i > 0$, we have to consider the probability that the maximum of the N_i's over all these wrong i.c.'s, denoted here by N_{iw}, is not very large.

$$P((N_{iw} < y) = \left(\sum_{k=0}^{y} \binom{N}{k} p^k (1-p)^{N-k}\right)^{(2^{d_i}-2)} \approx \Phi\left(\frac{2y - N}{\sqrt{N}}\right)^{(2^{d_i}-2)}$$

for $y = 0, 1, \ldots, N$. It follows that,

$$\begin{aligned} P(i\text{th input is correctly identified}) &= P((N_{iw} < N_{ic}) \\ &= \sum_{y=0}^{N} \binom{N}{y} P((N_{iw} < y) p_i^y (1-p_i)^{N-y} \\ &= \sum_{y=0}^{N} \binom{N}{y} p_i^y (1-p_i)^{N-y} \left\{\Phi\left(\frac{2y-N}{\sqrt{N}}\right)\right\}^{(2^{d_i}-2)} \end{aligned}$$

It can be easily seen that the above calculations go through when $\delta_i < 0$.

The probabilities of correct identification of all the input i.c.s have to be multiplied in order to obtain the overall probability of correct identification. Figure 1 shows plots of the variation of cipherlength with the desired probability of correct identification of the initial condition of an LFSR having length 12 and 16, using both the methods for $\delta = 0.02$. Figure 2 shows the corresponding plots for $\delta = 0.06$. Note that,

- The cipherlength requirement to achieve a desired probability of correct identification, increases with increase in the size of the LFSR.
- As δ (the separation from 0.5) increases, the cipherlength requirement for a fixed probability of correct identification, reduces remarkably.

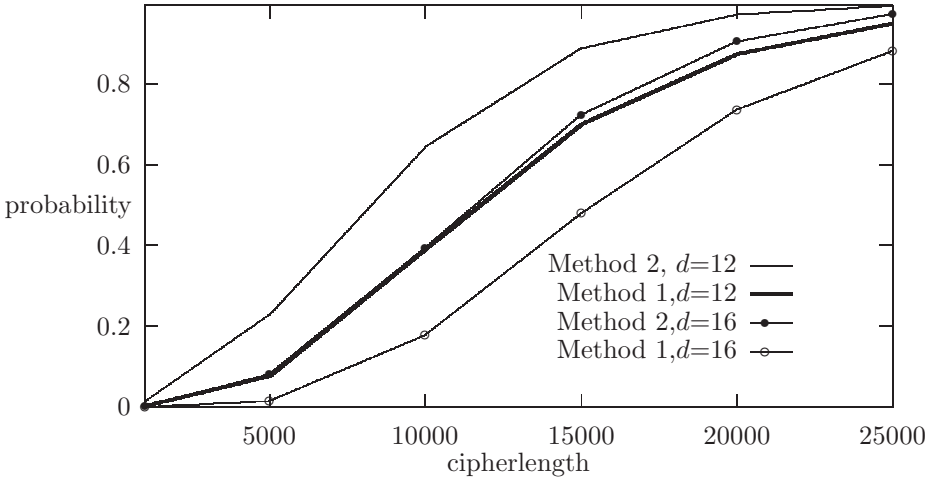

Figure 1: Probability of correct identification of an i.c. vs. cipherlength required, $\delta = 0.02$

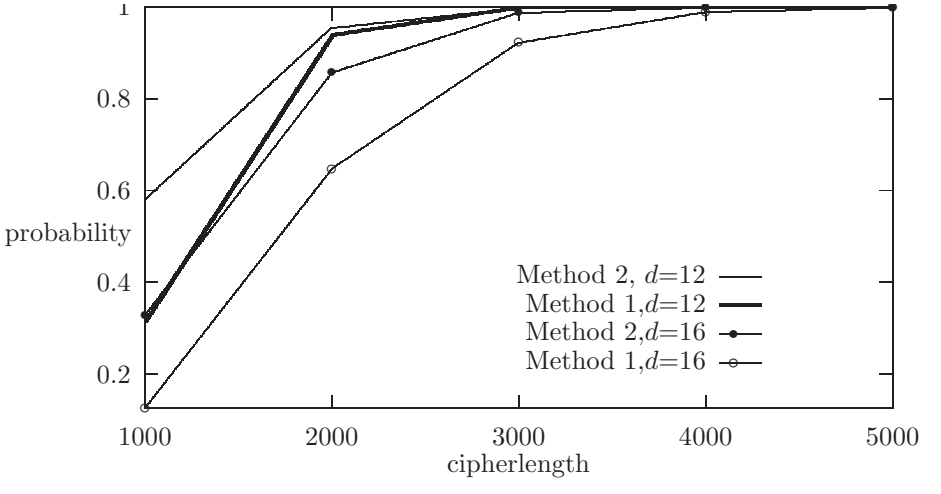

Figure 2: Probability of correct identification of an i.c. vs. cipherlength required, $\delta = 0.06$

It can also be seen that the cipherlength requirements for Method 2 are much less than that of Method 1. For the rest of the paper, only Method 2, *i.e.* the 'best-is-correct' approach shall be adopted.

3 Estimation of the Combining Function

We now consider the case where the combining function is unknown, but the LFSR i.c.'s have been correctly identified. This would help us eventually in addressing the problem of identifying the initial conditions and the combining function when both are unknown. The latter case is considered in Section 4.

3.1 A Maximum Likelihood Approach

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.

Suppose that p_0 is the probability that a bit of the plain text stream is equal to 0. As mentioned earlier, for all practical coding schemes, $p_0 \neq 0.5$. Throughout this paper, we have used the Murray code, for which $p_0 = 0.58$. Given that the corresponding function output is Y_j, the ith bit C_i of the cipher stream 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

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

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 cipherstream given the input streams is

$$\prod_{j=0}^{2^m-1} \prod_{i \in I_j} P(C_i = c_i|Y_j) = \prod_{j=0}^{2^m-1} \prod_{i \in I_j} [1 - p_0 + (2p_0 - 1)Y_j]^{c_i} \cdot [p_0 - (2p_0 - 1)Y_j]^{1-c_i}.$$

Thus, the likelihood of $Y_0, Y_1, \ldots, Y_{2^m-1}$ given the input streams and the cipherstream 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

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

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.

Thus, the algorithm for function estimation can be summarized in the following steps:

1. Select a particular input combination
2. Count the number of times $Y = 1$ for a particular input combination. Count the number of times this particular input combination occurs.
3. Compute the proportion of 1's from the above counts.
4. Conclude that output (Y) for that input combination is 0 if proportion of 1's is less than 0.5 , and 1 otherwise.
5. Repeat steps 1-4 for another input combination until all combinations are exhausted.

3.2 Cipher Length Requirements

Suppose that for the jth input combination, the function output $Y_j = 0$. Assume that the index set for this input combination is I_j (with N_j elements). Then Y_j is correctly identified if at most $N_j/2$ out of the N_j cipher bits corresponding to the index set I_j turn out to be 1. This in turn occurs when at most $N_j/2$ of the corresponding N_j bits of the plain text happen to be 1. The latter event has probability $\sum_{k=N_j/2}^{N_j} \binom{N_j}{k} p_0^k (1 - p_0)^{N_j - k}$. It is easy to see that the above expression for the probability of correct identification of Y_j holds even when $Y_j = 1$.

Using the fact that the numbers $N_0, N_1, \ldots, N_{2^m - 1}$ have a multinomial distribution, we have

$$P(\text{The entire } m\text{-input truth table is correctly identified})$$
$$= \sum_{N_0,N_1,\ldots,N_{2^m-1}} \frac{N!}{N_0!N_1!\ldots N_{2^m-1}!}\left(\frac{1}{2^m}\right)^N \prod_{j=0}^{2^m-1}\left(\sum_{k=N_j/2}^{N_j}\binom{N_j}{k}p_0^k(1-p_0)^{N_j-k}\right)$$

The actual computation of this quantity can be extremely time-consuming, because of the multiple summations. Alternatively, one can ignore the dependence of the cell frequencies, and use a binomial distribution for each N_j, with probability of inclusion $1/2^m$. The (ignored) correlation of two cell frequencies is 2^{-2m}, which is small for $m \geq 3$. This leads to the following approximation:

$$P(\text{The entire } m\text{-input truth table is correctly identified})$$
$$= \left\{\sum_{k=0}^{N}\left(\binom{N}{k}\left(\frac{1}{2^m}\right)^k\left(1-\frac{1}{2^m}\right)^{N-k}\left(\sum_{l=[k/2]}^{k}\binom{k}{l}p^l(1-p)^{k-l}\right)\right)\right\}^{2^m} \quad (2)$$

Figure 3 shows the cipher length requirements vs. the probability of correct estimation of the function for a three-input and a five-input function. As expected, the cipher length required to achieve a particular probability of estimation is much more in the case of a five-input function than a three-input one.

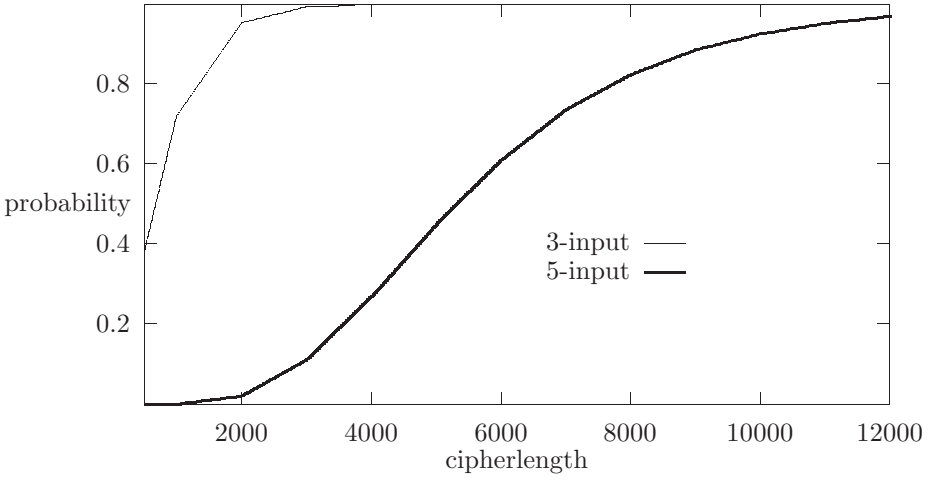

Figure 3: Probability of estimation of a function vs. cipherlength required

4 Determination of the Initial Conditions when the Combining Function Is Unknown

Let us assume that the combining function is not correlation immune with respect to any of its inputs. If the function and the i.c.s are unknown, the quantity $f_i - 0.5$ though still non-zero for the correct initial condition, may be either positive or negative. Hence, the 'best-is-correct' approach is slightly modified as follows. Check for the maximum of $|f_i - 0.5|$ over all possible initial conditions. Identify the maximiser as the correct i.c. of the ith input. Repeat for the other inputs. Once all the i.c.s are identified, estimate the function as outlined in the previous section.

Correspondingly, we can compute $P(i\text{th initial condition is correctly identified})$

$$
\begin{aligned}
&= \sum_{y=0}^{N/2} \binom{N}{y} p_i^y (1-p_i)^{N-y} \left(\sum_{k=y}^{N-y} \binom{N}{k} (0.5)^k (0.5)^{N-k} \right)^{(2^{d_i}-2)} \\
&\quad + \sum_{y=N/2+1}^{N} \binom{N}{y} p_i^y (1-p_i)^{N-y} \left(\sum_{k=N-y}^{y} \binom{N}{k} (0.5)^k (0.5)^{N-k} \right)^{(2^{d_i}-2)} \\
&\approx \sum_{y=0}^{N} \binom{N}{y} p_i^y (1-p_i)^{N-y} \left[2\Phi\left(\frac{|N-2y|}{\sqrt{N}} \right) - 1 \right]^{2^{d_i}-2}
\end{aligned}
$$

So, $P(\text{all i.c.s are correctly identified})$,

$$
P_{ic} = \prod_{i=1}^{m} \left[\sum_{y=0}^{N} \binom{N}{y} p_i^y (1-p_i)^{N-y} \left\{ 2\Phi\left(\frac{|N-2y|}{\sqrt{N}} \right) - 1 \right\}^{2^{d_i}-2} \right] \tag{3}
$$

$$
\text{Hence} \quad P(\text{all i.c.s and function are correctly identified}) = P_f P_{ic} \tag{4}
$$

where P_f is the probability of correct estimation of the function given that the i.c.s have been correctly determined, and is equal to the expression in (2)).

In order to give an idea of the cipher length requirements involved, we consider the estimation of the inputs for the case:
$f = X_1X_2 + X_3$
where the LFSR of each input is of size 12. The values $p_1 = 0.52$, $p_2 = 0.52$, $p_3 = 0.56$ are computed from the truthtable. We set the overall probability of correct identification equal to 0.95. Using (3) we obtain a cipherlength of 20,225 bits for the identification of the 3 initial conditions. On the other hand, using (4), we obtain a cipherlength of 21,450 bits for the identification of the initial conditions as well as estimation of the functions. It is indeed interesting to note that the task of function estimation requires very little of additional cipherlength over that of the job of identification of the initial conditions alone.

5 Determination of Initial Conditions when the Function Is Correlation Immune

Correlation immunity of a function from an information theoretic viewpoint has been described by Siegenthaler in [8]. An equivalent definition, given in [10] is as follows: An m-variable function $Y = f(X_1, X_2, \cdots, X_m)$ is lth order correlation immune iff

$$P(Y = X_{i_k} | X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_{k-1}} = 0) = 0.5,$$

where $i_1, i_2, \ldots, i_k$ is a set of distinct indices between 1 and m, $1 \leq k \leq l$.

5.1 Method

It follows the above definition of correlation immunity that for an m-variable, lth order correlation immune function, there is an input X_i and an index set $\{i_1, i_2, \ldots, i_{l+1}\}$ such that

$$P(Y = X_{i_{l+1}} | X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_l} = 0) \neq 0.5.$$

Further,

$$P(C = X_{i_{l+1}} | X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_l} = 0) \begin{cases} \neq 0.5 & \text{for correct i.c.,} \\ = 0.5 & \text{for wrong i.c.} \end{cases} \quad (5)$$

Note that the conditioning on the *specific* combination $X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_l} = 0$ is done without loss of generality. It was shown by [10] that conditioning on any other combination of values of the same inputs would produce a probability away from 0.5 when the correct initial conditions have been chosen.

Based on this idea, we adopt the following approach for determining the initial conditions when the combining function is unknown and correlation immune of an unknown order:

1. For the ith input, compute empirically $P(C = X_i | X_j = 0) = f_{ij}$, $i \neq j$, $1 \leq i, j \leq m$, for all possible initial condition pairs. If the maximum of $|f_{ij} - 0.5|$ is reasonably large as well as sufficiently separated from the next (lower) value of $|f_{ij} - 0.5|$, then it can be safely deduced that the corresponding initial conditions for i and j are the required initial conditions. Continue the procedure for all i-j combinations. Stop, if this results in the determination of all the initial conditions, otherwise proceed to Step 2.
2. For the ith input, compute empirically $P(C = X_i | X_j = 0, X_k = 0) = f_{ijk}$, $i \neq j \neq k$, $1 \leq i, j, k \leq m$ for all possible input combinations triplets. As before, if the maximum of $|f_{ij} - 0.5|$ is reasonably large and well separated from the rest, then the corresponding initial conditions are the correct ones. Continue this procedure for all i, j, k combinations. Stop, if all initial conditions have been determined. Otherwise, proceed to Step 3.
3. Condition on three inputs and proceed as before. If this fails, condition on four inputs, and so on.

5.2 Analysis

Let us use the notation $p_{i_{l+1}|i_1 \cdots i_l}$ for $P(Y = X_{i_{l+1}}|X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_l} = 0)$, when the correct initial condition has been used. Note that the number of wrong i.c.s in this case is $\prod_{j=1}^{l+1}(2^{d_j} - 2)$. Therefore, the probability of correct identification of the i.c.s of $i_1, i_2, \ldots, i_{l+1}$, after conditioning $X_{i_{l+1}}$ on $X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_l} = 0)$, has an expression similar to that of P_{ic} in (3):

$$\sum_{y=0}^{n} \binom{n}{y} p_{i_{l+1}|i_1 \cdots i_l}^{y} (1 - p_{i_{l+1}|i_1 \cdots i_l})^{n-y} \left\{ 2\Phi\left(\frac{|n-2y|}{\sqrt{n}}\right) - 1 \right\}^{\prod_{j=1}^{l+1}(2^{d_j}-2)},$$

where n is the number of bits (out of N) for which the input combination $X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_l} = 0)$ actually occurs. It is clear that n has a Binomial distribution, $Bin(N, 2^{-l})$. Therefore, the probability of correct identification of the i.c.s of $i_1, i_2, \ldots, i_{l+1}$, after conditioning $X_{i_{l+1}}$ on $X_{i_1} = 0, X_{i_2} = 0, \cdots, X_{i_l} = 0)$ is

$$P_{i_{l+1}|i_1 \cdots i_l} = \sum_{n=0}^{N} \binom{N}{n} (2^{-ln})(1 - 2^{-l})^{N-n} \sum_{y=0}^{n} \binom{n}{y} p_{i_{l+1}|i_1 \cdots i_l}^{y} (1 - p_{i_{l+1}|i_1 \cdots i_l})^{n-y} \left\{ 2\Phi\left(\frac{|n-2y|}{\sqrt{n}}\right) - 1 \right\}^{\prod_{j=1}^{l+1}(2^{d_j}-2)}. \quad (6)$$

5.3 An Example

We illustrate the method for the function $X_1 + X_2 + X_4 + X_3X_5 + X_4X_5$, which is second-order correlation immune, and shift register sizes 4,5,6,7,8 respectively. The cipher length is taken as 24,000. Here, $P(C = X_1|X_2, X_3 = 0) = P(C = X_4|X_1, X_2 = 0) = .54$. The input X_5 is third order correlation immune, and $P(C = X_5|X_1 = X_2 = X_3 = 0) = .46$. We start with the assumption that the function is unknown.

The empirical values of $P(C = X_i)$ are first calculated but none observed to be significantly away from 0.5, specifically, the maximum separation was 0.01. Next, the empirical values of $P(C = X_{i_k}|X_{i_j} = 0)$ for all possible but distinct values of i and j are calculated. Once again, these are all close to 0.5, with a maximum deviation of 0.02. After this, the empirical values of $P(C = X_{i_k}|X_{i_j} = X_{i_l} = 0)$ are calculated. It is observed that the empirical version of $P(C = X_1|X_2 = X_3 = 0)$, has the largest separation (0.048) from 0.5 for the correct combination of i.c.s of X_1, X_2 and X_3, while the second highest separation is 0.037. Using the identified i.c.s of X_1 and X_2, the empirical value of $P(C = X_4|X_1 = X_2 = 0)$ is also found to be furthest away from 0.5, specifically 0.044, when the correct i.c. for X_4 is used. The input X_5 however, is found to have 'input immunity' of order 3 *i.e.*, conditioning on two inputs does not yield a large enough value for the corresponding fraction of coincidence. (We define input immunity as follows: An input i has immunity of order m if $P(C = X_i|X_{i_1}, X_{i_2}, \ldots, X_{i_{m-1}} = 0) = 1/2$, $i \neq \{i_1, \ldots i_{m-1}\}$). It is found that

the empirical value of $P(C = X_5|X_1 = X_2 = X_3 = 0)$ has a well- defined separation from 0.5 (equal to 0.047), for the correct i.c. for X_5. For every wrong initial condition of X_5 the separation from 0.5 is much less with a maximum equal to 0.037.

5.4 The 'Diameter' Approach

Note from (5) that the conditional probability of coincidence is only known to be 'away from 0.5'. The precise value of this probability may depend on the combination of values of the inputs on which the conditioning is made. Specifically, the 'all zero' combination may not produce a conditional probability *furthest* away from 0.5. In order to extract the maximum possible information from the conditioning process, we may try and condition on all possible combination of values of the conditioning inputs, and look for the maximum deviation from 0.5. Thus, for every input combination $X_{i_1}, X_{i_2}, \ldots, X_{i_{l+1}}$ we consider the absolute difference

$$\max_{\substack{u_1,u_2,\ldots,u_l \text{ binary} \\ v_1,v_2,\ldots,v_l \text{ binary}}} \Big| P(C = X_{i_{l+1}}|X_{i_1} = u_1, X_{i_2} = u_2, \cdots, X_{i_l} = u_l) - P(C = X_{i_{l+1}}|X_{i_1} = v_1, X_{i_2} = v_2, \cdots, X_{i_l} = v_l) \Big| \tag{7}$$

which should be far away from 0 for the correct i.c. and equal to 0 for any wrong i.c. The empirical version of this maximum difference can be computed using the observed fraction of coincidences, and used in the procedure given in Section 5.3 in lieu of $|f_{ij} - 0.5|$. The maximum absolute difference can be thought of as the diameter of the set of all candidate fractions. Therefore, we call this approach the 'diameter' approach.

When the diameter approach is used in the example of the above Section, it is found that the i.c.s for X_1, X_2 and X_3 are correctly determined for a cipher length of only 16,000 bits. The corresponding fraction of coincidence (best) is 0.093 while the one immediately next to it in magnitude is 0.052. Thus, we see that the 'best' diameter is well separated from both 0 as well as the 'next best' diameter for even 16,000 bits. The rest of the search for the i.c.s of X_4 and X_5 are carried out successfully in a similar way. This suggests that a reasonable reduction in the cipher length requirements may be obtained using the 'diameter' approach.

6 Computational Work

The size of the space of test initial conditions grows exponentially with increase in the size of the LFSRs. For an input of size d_i which is not correlation immune, the search time is proportional to 2^{d_i}. If the input 1 has input immunity of order m and $d_2, \ldots, d_m$ are corresponding conditioning inputs, then the search time is proportional to $2^{\sum d_i} = 2^{d_1+d_2+\cdots+d_m}$. The overall computational time is of the

order of the time needed for the determining the input with largest immunity and the corresponding conditioning inputs. Estimation of the combining function would have no effect on the order of the search time.

The actual software implementation of the LFSR simulation and output generation followed by comparison with the cipherstream is done based on certain efficient features proposed in [11]. To begin with, the cipherstream is 'packed' into an array. This is done by moving blocks of the ciphertext data (of length 16, in this paper) into each location of the array. For example if `ctext[ ]` represents the packed ciphertext array and $\{c_0, c_1, c_2, \ldots, c_N\}$ is the ciphertext, then `ctext[0]` holds $c_0, c_1, \ldots, c_{15}$, `ctext[1]` holds $c_{16}, c_{17}, \ldots, c_{31}$ and so on. Next, the LFSR output is generated in packed form. The content of each packed location is compared with the content of the corresponding packed ciphertext array location by bitwise ex-oring. The number of bits for which the two do not match (the number of *ones* in the bit configuration of the resulting number) is read from a table-lookup. This number is obtained cumulatively for the entire packed ciphertext array. From this, the relevant fraction of coincidences is computed.

It was experimentally observed that for a LFSR of length 16, determination of the correct i.c. using no 'packing' required 985 seconds on a 333 MHz Pentium Processor. On the other hand, the algorithm incorporating data packing required only 89 seconds implying a significant speedup. Modifications for further increase in speed of the algorithm are currently being explored.

In order to have an idea of the actual running times involved, the algorithm with data packing was used on a 333 MHz Pentium processor. Determining the initial conditions for an input with input immunity 2,$\{d_1 = 4, d_2 = 5, d_3 = 6\}$ required 900 seconds, $\{d_1 = 4, d_2 = 5, d_3 = 7\}$ required 1908 seconds while $\{d_1 = 5, d_2 = 6, d_3 = 7\}$ required 8429 seconds.

7 Conclusions

We have shown in Section 4 that the knowledge of the combining function is not a very important one, because the lack of this knowledge entails minimal increase in the cipherlength needed to break the code. The assumption of known shift register sizes and polynomials is easily removed if one permits a larger search space, *i.e.* the search must now include varying shift register sizes and polynomials. Other modifications such as parallelization must be explored to reduce the resulting computational workload.

In order to reduce the cipher length requirements one may modify the 'best-is-correct' approach to include a reasonable size of candidate solutions ($k > 0$). For example, the best 5% of the fractions of coincidence may be chosen and the corresponding i.c.s used to decrypt the message. The correct i.c.s will be the ones corresponding to which meaningful text (English or otherwise) is generated. This generation may even be automatised using some prominent features of the language. The trade-off between cipher length requirements and the computation needed for automatic checking of trial solutions should be an interesting subject of further study.

The primary bottleneck of the approach developed for correlation immune combining functions is the tremendous size of the search space for even low sizes of the LFSRs. The use of even fast modifications of the basic algorithm [11] do not contribute much to decreasing the computation time. Hence, in order to make such an encryption system cryptologically strong, the designer has to choose a combining function having a large number of inputs and correlation immunity of sufficiently high order (not too high because then, search by enumeration would be possible).

References

1. A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997. 306
2. T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Transactions on Computers*, Vol. c-34, No.1, January 1985, pp. 81-85. 306, 307, 308
3. K. Zeng and M. Huang, "On the linear syndrome method in cryptanalysis," *Advances in Cryptology - CRYPTO '88*, Vol. 403, S. Goldwasser, editor, Springer Verlag 1990, pp. 469-478. 307
4. V. Chepyzhov and B. Smeets, "On a fast correlation attack on stream ciphers," *Advances in Cryptology - EUROCRYPT '91*, Vol. 547, D.W. Davies, editor, Springer Verlag, 1991, pp. 176-185. 307
5. A. Clark, J. Golić and E. Dawson, "A comparison of fast correlation attacks," *Fast Software Encryption*, Third International Workshop (LNCS 1039), D. Gollman, editor, Springer Verlag, 1996, pp. 145-157. 307
6. R. Forre, "A fast correlation attack on non-linearly feedforward filtered shift register sequences," *Advances in Cryptology - EUROCRYPT '89* (LNCS 434), 1990, pp. 586-95. 307
7. M.J. Mihaljevic and J. Golić, "A comparison of cryptanalytic principles based on iterative error-correction," *Advances in Cryptology - EUROCRYPT '91*, Vol. 547, D.W. Davies, editor, Springer Verlag 1991, pp. 527-531. 307
8. T. Siegenthaler, "Correlation-Immunity of Nonlinear Combining functions for Cryptographic Applications," *IEEE Transactions on Information Theory*, Vol. 30, No. 5, September 1984, pp.776 - 780.
9. B.K. Roy, "Ciphertext only cryptanalysis of LFSR based encryption schemes," *Proceedings of the National Seminar on Cryptology*, Delhi, July 1998, pp.A-19–A-24. 307, 316 308
10. S. Maitra, P. Sarkar and B.K. Roy, "A new definition of correlation immunity of Boolean functions," *Technical Report No. ASD/98/08* Indian Statistical Institute, June 1998. 316
11. S. Maitra and P. Sarkar, "Efficient implementation of ciphertext only attack on LFSR based encryption schemes," *Proceedings of the National Seminar on cryptology*, Delhi, July 1998, pp. A-1–A-12. 319, 320