A Comparison of Fast Correlation Attacks

Andrew Clark[1], Jovan Dj. Golić[1,2] * and Ed Dawson[1]

[1] Information Security Research Centre, Queensland University of Technology
GPO Box 2434, Brisbane Q 4001, Australia
[2] School of Electrical Engineering, University of Belgrade
Email: {*aclark,golic,dawson*}*@fit.qut.edu.au*

Abstract. A comparison of two different approaches to fast correlation attacks on stream ciphers is conducted. One is based on standard or modified iterative probabilistic decoding algorithms, and the other is a recent, so-called free energy minimisation approach. Two different comparisons are presented: one based on the Hamming distance and the other on error-free information sets. The results indicate that a modified iterative probabilistic decoding attack outperforms the free energy minimisation attack in high noise probability regions.

1 Introduction

The concept of a fast correlation attack was first introduced in [8] where a bit-by-bit reconstruction procedure based on iterative probabilistic and threshold decoding is proposed (see also [13]). The underlying ideas regarding the probabilistic decoding of low density parity-check linear codes can be found in [3], [7] and [2]. After the pioneering work of Meier and Staffelbach [8] and Zeng and Huang [13], various algorithms of this type have been published and theoretically or experimentally analysed (for example, see [1], [9, 10, 11], [14] and [5]). All of them share two basic features, an iterative error-correction algorithm and a method of obtaining low density parity-checks.

In this paper a modified version [10, 4] of the Meier-Staffelbach attack [8] is considered which we label the *iterative probabilistic decoding* (IPD) approach. A simple modification [4] to the IPD attack based on *fast resetting* is shown to produce improved results. This is called the *modified* IPD (or MIPD) attack. In [6] MacKay proposed a new, interesting approach to fast correlation attacks using a *free energy minimisation* (FEM) algorithm. This algorithm gives an improvement over the basic IPD attack.

In Section 2 a basic iterative probabilistic decoding algorithm is defined which is a variation on the original Meier-Staffelbach attack. The fast resetting modification to this algorithm is also described in Section 2. Section 3 introduces the free energy minimisation algorithm as proposed by MacKay. In Section 4 the experimental results and a comparison of the algorithms are given. Two different comparisons are applied: the first is based on the Hamming distance and the

* This research was supported in part by the Science Fund of Serbia, Grant #0403, through the Institute of Mathematics, Serbian Academy of Arts and Sciences.

second makes use of error-free information sets [2]. Recall that an information set is any set of linearly independent bits in a LFSR sequence, from which one can reconstruct the whole sequence. A comparison based simply on whether or not the algorithm successfully recovers the entire LFSR sequence [6] may be misleading. One set of results is based on the minimum Hamming distance between the recovered sequence and the actual LFSR output stream. The second set of results presented here make use of a sliding window technique [9] which searches for an information set containing r consecutive bits (where r is the length of the shift register). This is especially useful when considering the high noise probability region where some, but not all, of the LFSR sequence has been successfully reconstructed. The experimental results show that the MIPD algorithm is superior to the FEM algorithm [6]. In Section 5 a summary of results and conclusions are presented.

2 Iterative Probabilistic Decoding Algorithm

As Fig. 1 indicates, the observed keystream sequence $z = \{z_i\}_{i=1}^N$ is regarded as a noise-corrupted version of the LFSR sequence $a = \{a_i\}_{i=1}^N$, that is, $z_i = a_i \oplus e_i$, '$\oplus$' denoting the modulo 2 addition, where $e = \{e_i\}_{i=1}^N$ is a binary random noise sequence such that $\Pr\{e_i = 1\} = p$ for every $i = 1, \ldots, N$. This model [12] is called a *binary symmetric memoryless channel*. Without loss of generality, the correlation coefficient defined as $c = 1 - 2p$ is assumed to be positive. Let $f(x)$ be the characteristic polynomial of a LFSR of length r which is assumed to be known. The problem is to reconstruct the LFSR sequence, $a = \{a_i\}_{i=1}^N$, from the observed keystream sequence, $z = \{z_i\}_{i=1}^N$, where $r < N < 2^r - 1$ and the value of N should be as small as possible.

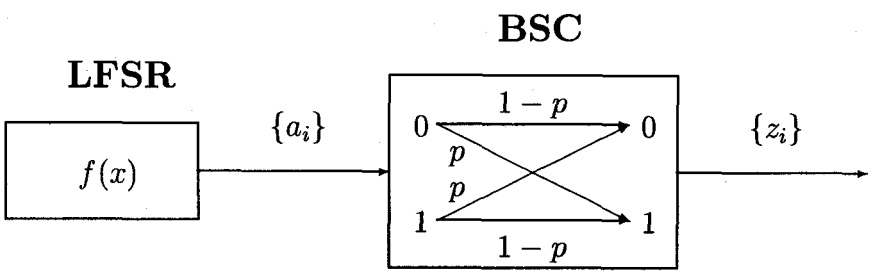

Fig. 1. Model of a stream cipher as used in the correlation attack.

A *parity-check* is any linear relationship satisfied by a LFSR sequence. It is well known that the parity-checks correspond to polynomial multiples of $f(x)$ (see [1]). In fact, any polynomial multiple whose weight (the number of nonzero

terms) equals W defines a set of W parity-checks corresponding to its different phase shifts. In [8] a simple algebraic technique is applied to derive a set of parity-checks. This technique involves repeated squaring of the characteristic polynomial $f(x)$ until a polynomial is found with the maximum degree not greater than $N-1$. Each corresponding parity-check then involves the same number of terms, i.e., has the same weight $w+1$, where w is the number of taps. A set of parity-checks is called orthogonal on a given bit if, except for that bit, every other involved bit appears in exactly one of the parity-checks.

Let $\Pi_i = \{\pi_k(i)\}_{k=1}^{|\Pi_i|}$ ($|\Pi_i|$ denotes the cardinality of Π_i) be a set of parity-checks orthogonal on the ith bit, $i = 1, \ldots, N$. Let a parity-check value be defined as the modulo 2 sum $c_k(i) = \sum_{l \in \pi_k(i)} z_l$. Since the noise sequence e is random, so are the parity-check values. When the parity-checks are orthogonal, the corresponding posterior probability for noise bits given the parity-check values is then given as [10]

$$p_i = \Pr(e_i = 1 | \{c_k(i)\}_{k=1}^{|\Pi_i|}) \tag{1}$$

$$= \frac{q_i \prod_{k=1}^{|\Pi_i|} q_k(i)^{\bar{c}_k(i)} (1 - q_k(i))^{c_k(i)}}{q_i \prod_{k=1}^{|\Pi_i|} q_k(i)^{\bar{c}_k(i)} (1 - q_k(i))^{c_k(i)} + (1 - q_i) \prod_{k=1}^{|\Pi_i|} (1 - q_k(i))^{\bar{c}_k(i)} q_k(i)^{c_k(i)}}$$

for any $i = 1, \ldots, N$, where q_i denotes the prior probability for the ith noise bit to be equal to one, $\bar{c}_k(i) = 1 - c_k(i)$, $q_k(i) = (1 - \prod_{l=1}^{w}(1 - 2q_{m_l}))/2$ and $\{m_l\}_{l=1}^{w}$ denotes the set of indices of the bits involved in the parity-check $\pi_k(i)$, for any $k = 1, \ldots, |\Pi_i|$ and $i = 1, \ldots, N$. Initially, the prior probabilities for noise bits are all set to p. In the succeeding iterations, the posterior probabilities from the preceding iteration are used as the prior probabilities for the current one. The same expression (1) can be used even if the parity-checks are not orthogonal. An analogous expression can be derived for the parity-checks of different weights [11].

2.1 Basic IPD Algorithm

In this section we consider a decoding algorithm given in [10] which is an iterative procedure employing the parity-checks of possibly different weights and a Bayesian decision rule in error-correction for each bit. Each iteration consists of two main stages. In the first stage the parity-checks are calculated bit-by-bit. In the second stage, the Bayesian bit-by-bit error-correction is made based on the estimation of the relevant posterior probabilities obtained by using the posterior probabilities from the preceding iteration as the prior probabilities in the current iteration, and in the first iteration the prior probabilities are all set to p. Namely, if the posterior probability p_i is greater than one half, then the noise bit e_i is set to one, the keystream bit z_i is complemented and p_i is set to $1 - p_i$.

In the first iteration the optimal Bayesian decision minimises the symbol error-rate, p_e. In the succeeding iterations the error-rate almost always decreases

for two reasons. Firstly, the error-correction performed by the algorithm may reduce the number of errors in the updated observed keystream sequence and, secondly, recycling the probability vector according to (1) causes the probabilities for most noise bits to converge to zero (after complementation). Often the algorithm becomes trapped in a local minimum of the error-rate, $p_e \approx 0$, after which the error-correction of the observed keystream sequence ceases. In this case (as was proposed in [8]), when the error-rate is less than ϵ, one can substitute the initial probability vector for the prior one and continue from step 1. This is called *resetting* the algorithm. Resetting the algorithm enhances the error-correction capability of the algorithm and increases the number of satisfied parity-checks. The set of iterations between two successive resets is called a *round*. Note that the information set decoding technique, when applied, is also used as a stopping criterion. The basic error-correcting algorithm is as follows.

Algorithm IPD

- *Input:* The observed keystream sequence $z = \{z_i\}_{i=1}^{N}$, p and a set of (orthogonal) parity-checks $\Pi_i = \{\pi_k(i)\}_{k=1}^{|\Pi_i|}$, $i = 1, \ldots, N$.
- *Initialization:* $j = 0$, $k = 0$ and $q_i = p, i = 1, \ldots, N$, where j is current iteration index and k is the current round index. Also define the maximum number of rounds k_{max}, the minimum error-rate ϵ and the maximum number of iterations without change in the number of satisfied parity-checks, J.
- *Resetting Criteria:* Probabilities are reset when the average error probability per symbol (error-rate) p_e drops below ϵ or when the number of satisfied parity-checks has not changed for J iterations.
- *Stopping Criteria:* The algorithm stops when the number of rounds reaches k_{max}, or, ideally, when all the parity-checks are satisfied.
- *Step 1:* Calculate the parity-check values $\{c_k(i)\}_{k=1}^{|\Pi_i|}$, $i = 1, \ldots, N$, on the observed keystream sequence. If all parity-checks are satisfied, go to step 7. If the number of satisfied parity-checks has not changed for J iterations, go to step 6.
- *Step 2:* Using equation (1) calculate posterior probabilities $p_i, i = 1, \ldots, N$.
- *Step 3:* If $p_i > 0.5$, set $z_i = z_i \oplus 1$, $p_i = 1 - p_i, i = 1, \ldots, N$.
- *Step 4:* Substitute the posterior probabilities of the current iteration for the prior probabilities of the next iteration: $q_i = p_i$, for $i = 1, \ldots, N$.
- *Step 5:* If $p_e = \frac{1}{N} \sum_{i=1}^{N} p_i > \epsilon$, increment j by 1 and go to step 1.
- *Step 6:* Set $q_i = p, i = 1, \ldots, N$, and increment k by 1. If $k < k_{max}$, go to step 1.
- *Step 7:* Set $a_i = z_i$, $i = 1, \ldots, N$, and stop the procedure.
- *Output:* The reconstructed LFSR sequence is $\{a_i\}_{i=1}^{N} = \{z_i\}_{i=1}^{N}$.

2.2 Modified IPD Algorithm with Fast Resetting

The underlying idea of resetting is explained above. By modifying this idea [4] we have improved the power of resetting which has resulted in increased number of satisfied parity-checks. The resetting used in step 4 of the basic algorithm is

called *slow resetting*. We introduce another type of resetting which is called *fast resetting*. Fast resetting is defined in the modified step 5 below.

Reconstruction of the observed keystream sequence is performed in step 3 of the basic algorithm when $z_i = z_i \oplus 1$. However, the complementations are not effective due to the posterior probability transformation $p_i \rightarrow 1 - p_i$ (recycling with or without complementations is all the same for orthogonal parity-checks, see [14]). They become effective only after resetting, where such a transformation does not take place. Of course, not all of these complementations are correct and the algorithm may introduce new errors to the observed keystream sequence. If the number of correct complementations exceeds the number of incorrect complementations, then the probability of error is reduced in the observed keystream sequence. This may not occur if we wait till the error-rate goes to zero due to the self-composition property of the basic algorithm. Accordingly, one may expect the performance to be improved if the resetting is done before the error-rate falls below a threshold. More precisely, when the cumulative number of complementations in each round reaches a predefined value, C, we substitute $q_i = p$, $i = 1, \ldots, N$. This is called fast resetting, and significantly improves the performance of the basic algorithm.

- *Step 5:* If $p_e = \frac{1}{n}\sum_{i=1}^{N} p_i > \epsilon$ and the cumulative number of complementations is less than C, increment j by 1 and go to step 1.

In practice, C can be optimised depending on the observed keystream sequence, the noise probability or the parity-checks used.

3 Free Energy Minimisation Algorithm

MacKay [6] presents an algorithm for solving a class of problems which can be represented in the form $(As + n) \bmod 2 = r$, where s is a binary vector of length N, n and r are binary vectors of length M and A is a binary matrix. The problem, then, is to solve for s given A and r. The algorithm is based on the 'variational free energy minimisation method'. For a discussion of this method the reader is referred to MacKay's paper [6]. When applied to a fast correlation attack, s denotes the unknown LFSR output sequence, r contains M parity-check values and n is essentially ignored (let it gradually approach 0). The matrix A has as its rows all the parity-checks for each bit position.

The problem is parameterized by a real valued vector θ of length N. Then q_n^1 and q_n^0 are defined as follows:

$$q_n^1 = \frac{1}{1 + e^{-\theta_n}} \tag{2}$$

$$q_n^0 = \frac{1}{1 + e^{+\theta_n}} \tag{3}$$

for $n = 1, \ldots, N$. It follows that q_n^1 and q_n^0 are related so that $\theta_n = \log(q_n^1/q_n^0)$. A constant

$$b = \log\left(\frac{p}{1-p}\right) \tag{4}$$

is called the *bias* constant, where p is the noise probability from Fig. 1. Let g be a $(1,-1)$ binary encoding of the parity-check vector (or syndrome vector) $r = Az$, obtained from the observed keystream sequence. Probabilities $p^1_{m,\nu}$ and $p^0_{m,\nu}$ are defined as the probability that the partial sum $\sum_{n=1}^{\nu} A_{m,n}s_n$ mod 2 is equal to 1 and 0, respectively. These probabilities

$$\left.\begin{aligned} p^1_{m,\nu} &= q^0_\nu p^1_{m,\nu-1} + q^1_\nu p^0_{m,\nu-1} \\ p^0_{m,\nu} &= q^0_\nu p^0_{m,\nu-1} + q^1_\nu p^1_{m,\nu-1} \end{aligned}\right\} \quad \text{if } A_{m,\nu} = 1$$

$$\left.\begin{aligned} p^1_{m,\nu} &= p^1_{m,\nu-1} \\ p^0_{m,\nu} &= p^0_{m,\nu-1} \end{aligned}\right\} \quad \text{if } A_{m,\nu} = 0 \tag{5}$$

have the initial condition $p^1_{m,0} = 0$ and $p^0_{m,0} = 1$. Similarly, let $r^1_{m,\nu}$ and $r^0_{m,\nu}$ be the probabilities that the partial sum $\sum_{n=\nu}^{N} A_{m,n}s_n$ mod 2 is equal to 1 and 0, respectively. They are obtained by an analogous reverse recursion of (5).

The free energy is broken up into three terms: *likelihood energy*, *prior energy* and *entropy*:

$$F(\theta) = E_L(\theta) + E_P(\theta) - S(\theta) \tag{6}$$

where

$$E_L(\theta) = -\sum_m g_m p^1_{m,N} \tag{7}$$

$$E_P(\theta) = -\sum_n b q^1_n \tag{8}$$

$$S(\theta) = -\sum_n \left(q^0_n \log q^0_n + q^1_n \log q^1_n\right). \tag{9}$$

It can be shown that the derivative of the free energy is given by:

$$\frac{\partial F}{\partial \theta_n} = q^1_n q^0_n \left(\theta_n - b - \sum_m g_m d_{m,n}\right) \tag{10}$$

where

$$\begin{aligned} d_{m,n} = {} & (p^1_{m,n-1} r^1_{m,n+1} + p^0_{m,n-1} r^0_{m,n+1}) - \\ & (p^1_{m,n-1} r^0_{m,n+1} + p^0_{m,n-1} r^1_{m,n+1}). \end{aligned} \tag{11}$$

By setting the derivative to zero a 're-estimation optimiser' is obtained which defines a recursive update procedure for the θ parameter. MacKay introduces an 'annealing' parameter β which gradually increases as the algorithm progresses. Its purpose is to prevent the search from heading too quickly into a local minimum. It should be noted that the annealing procedure is deterministic unlike some other annealing procedures (such as the simulated annealing approach [4]). The θ vector is then updated according to

$$\theta_n = b + \beta \sum_m g_m d_{m,n}, \quad n = 1, \ldots, N. \tag{12}$$

Algorithm FEM

- *Input:* Syndrome vector r, parity-check matrix A, p the noise probability, the initial value for β (β_0), the scaling factor for β (β_f), the maximum value for β (β_{max}) and the maximum number of iterations.
- *Initial Conditions:* Number of iterations = 0, $\beta = \beta_0$,

$$g_m = \begin{cases} +1 \text{ if } r_m = 1 \\ -1 \text{ if } r_m = 0 \end{cases}, \quad m = 1, \ldots, M$$

$$b = \theta_n = \log\left(\frac{p}{1-p}\right), \quad n = 1, \ldots, N$$

$$\left.\begin{matrix} p^1_{m,0} = r^1_{m,0} = 0 \\ p^0_{m,0} = r^0_{m,0} = 1 \end{matrix}\right\}, \quad m = 1, \ldots, M.$$

- *Stopping Condition:* Stop after a fixed number of iterations or when β exceeds β_{max}.
- *Step 1:* Update q^1_n, q^0_n, $n = 1, \ldots, N$, by using (2) and (3).
- *Step 2:* (Forward Pass) Update $p^1_{m,n}$ and $p^0_{m,n}$, $m = 1, \ldots, M$, and $n = 1, \ldots, N$, according to the recursion (5).
- *Step 3:* (Reverse Pass) Update $r^1_{m,n}$ and $r^0_{m,n}$ this time using (5) in the reverse direction, i.e., $m = M, \ldots, 1$ and $n = N, \ldots, 1$.
- *Step 4:* Update each $\theta_n, n = 1, \ldots, N$, by calculating the gradient and using equation (12).
- *Step 5:* Increment the number of iterations.
- *Step 6:* Calculate the free energy using (6). If the energy has decreased since the previous iteration return to step 1.
- *Step 7:* Scale β by β_f, i.e., let $\beta = \beta \times \beta_f$. If $\beta < \beta_{max}$ and the number of iterations is less than the maximum, then return to step 1.
- *Step 8:* Output the noise sequence as determined from θ as follows: if $\theta_n > 0$ output 1, otherwise output 0, for $n = 1, \ldots, N$. The LFSR sequence can be obtained as the modulo 2 sum of the output noise sequence and the observed keystream sequence.

4 Experimental Results and Comparison

In this section we present experimental results for the fast correlation attacks based on the IPD, MIPD and FEM algorithms. All results are averaged over 50 different noise samples. Shift registers using three different characteristic polynomials are used. In each case the number of taps is different (two, four and six). The chosen polynomials are all primitive and are given in

Number of taps	LFSR length	Primitive characteristic polynomial
2	31	$1+x^3+x^{31}$
4	50	$1+x^2+x^3+x^4+x^{50}$
6	72	$1+x+x^2+x^3+x^4+x^6+x^{72}$

For each trial, 10000 bits of observed keystream sequence are used. The characteristic polynomials were deliberately chosen to produce a set of non-orthogonal parity-checks (different phases are not orthogonal) to test the robustness of the attacks, since the IPD algorithm requires that the parity-checks be orthogonal while the FEM algorithm does not.

The best value for the variable threshold C in the fast resetting MIPD algorithm was determined experimentally. In the case of two taps, C was chosen to be 10. For the shift registers with four and six taps a value of 100 was used for C. For the FEM algorithm we used the values that are suggested in [6]: $\beta_0 = 0.25$, $\beta_f = 1.4$ and $\beta_{max} = 4$. Both algorithms used the same set of parity-checks for each of the respective characteristic polynomials. The parity-checks were obtained simply by repeated squaring of $f(x)$ until a polynomial with the maximum degree not exceeding $N-1$ (in this case 9999) is found. The number of parity-checks in each case is equal to $\lfloor \log_2(N-1)/r \rfloor + 1$ where r is the degree of $f(x)$. For shift registers of lengths 31, 50 and 72 the numbers of parity-checks are 9, 8 and 8, respectively. As suggested in [8], all the phases of the parity-checks are utilised. However, in the end regions only some of the phases can be used. In this case as many phases of the parity-checks as possible were used.

For each test, two sets of results were obtained. The first involved finding the minimum Hamming distance between the actual LFSR output and the solution that each algorithm found. This gives an indication of how close the algorithm is getting to the actual solution. The second test makes use of error-free information sets. A sliding window technique [9] is used in which a search for r consecutive bits satisfying the characteristic polynomial is made (r is the shift register length). If r such bits can be found, then the attack is deemed successful.

It is clear from each of Figures 2-4 that the fast resetting MIPD algorithm outperforms MacKay's FEM algorithm for each considered number of taps. It can also be seen that the FEM algorithm performs better than the slow resetting IPD algorithm if the number of taps increases. The two testing techniques, minimum Hamming distance and error-free information sets, appear to correlate

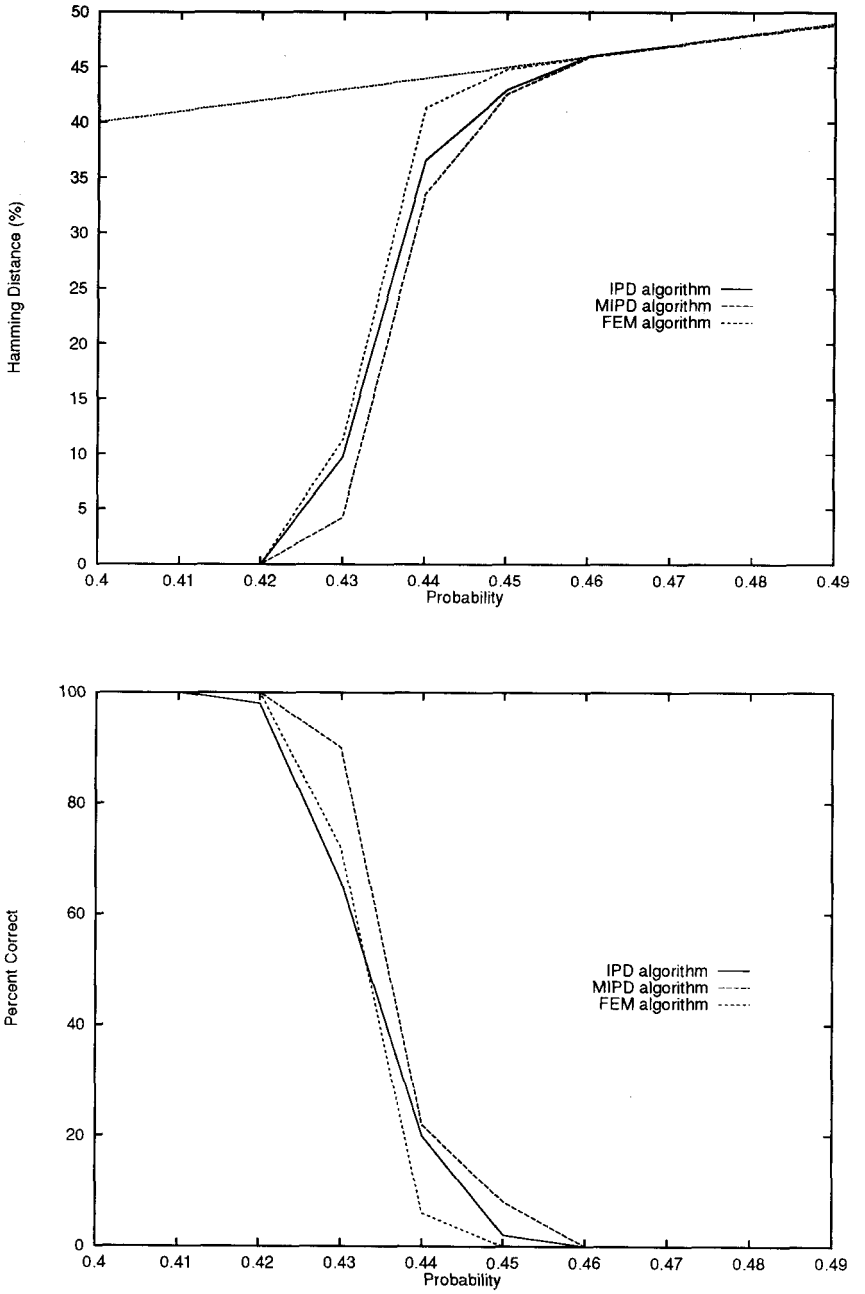

Fig. 2. Results for a shift register with 2 taps.

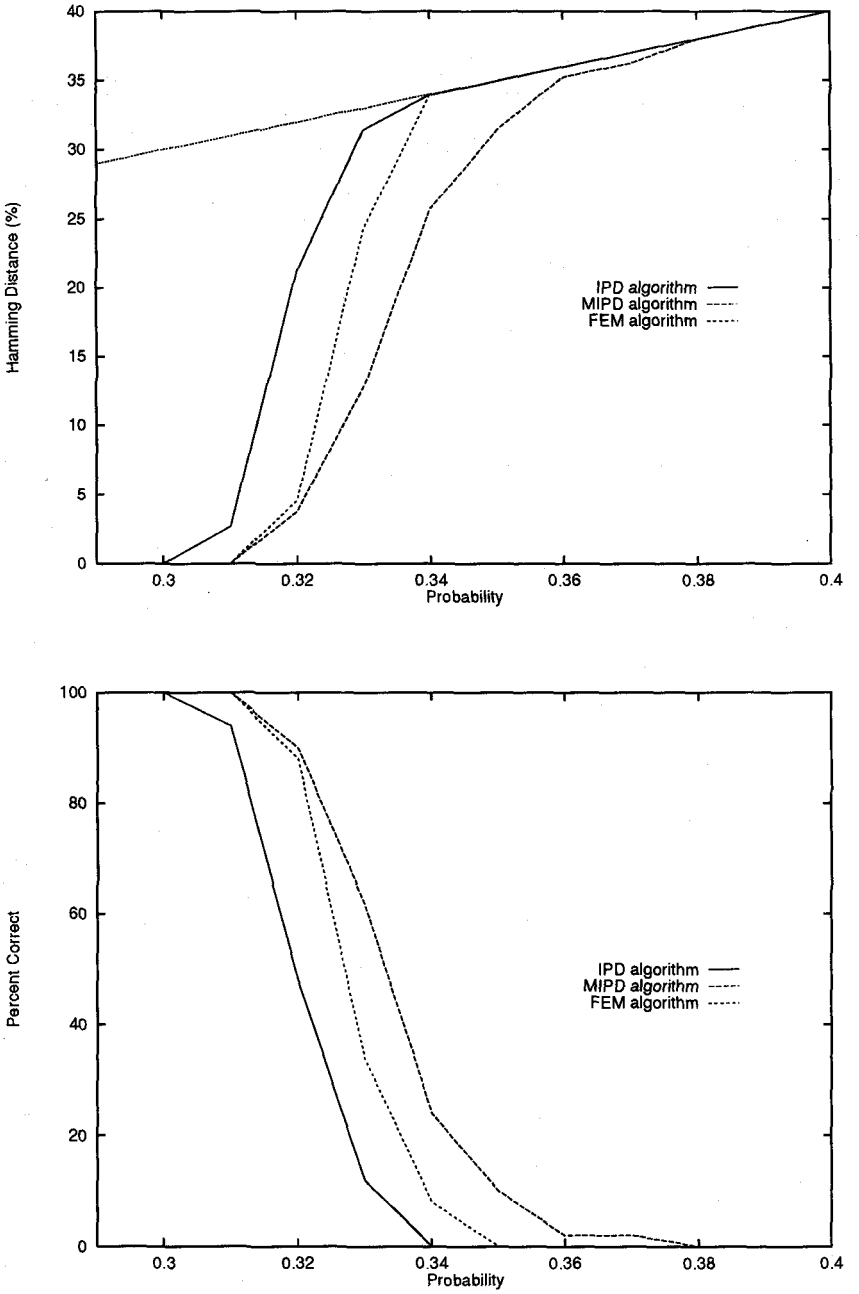

Fig. 3. Results for a shift register with 4 taps.

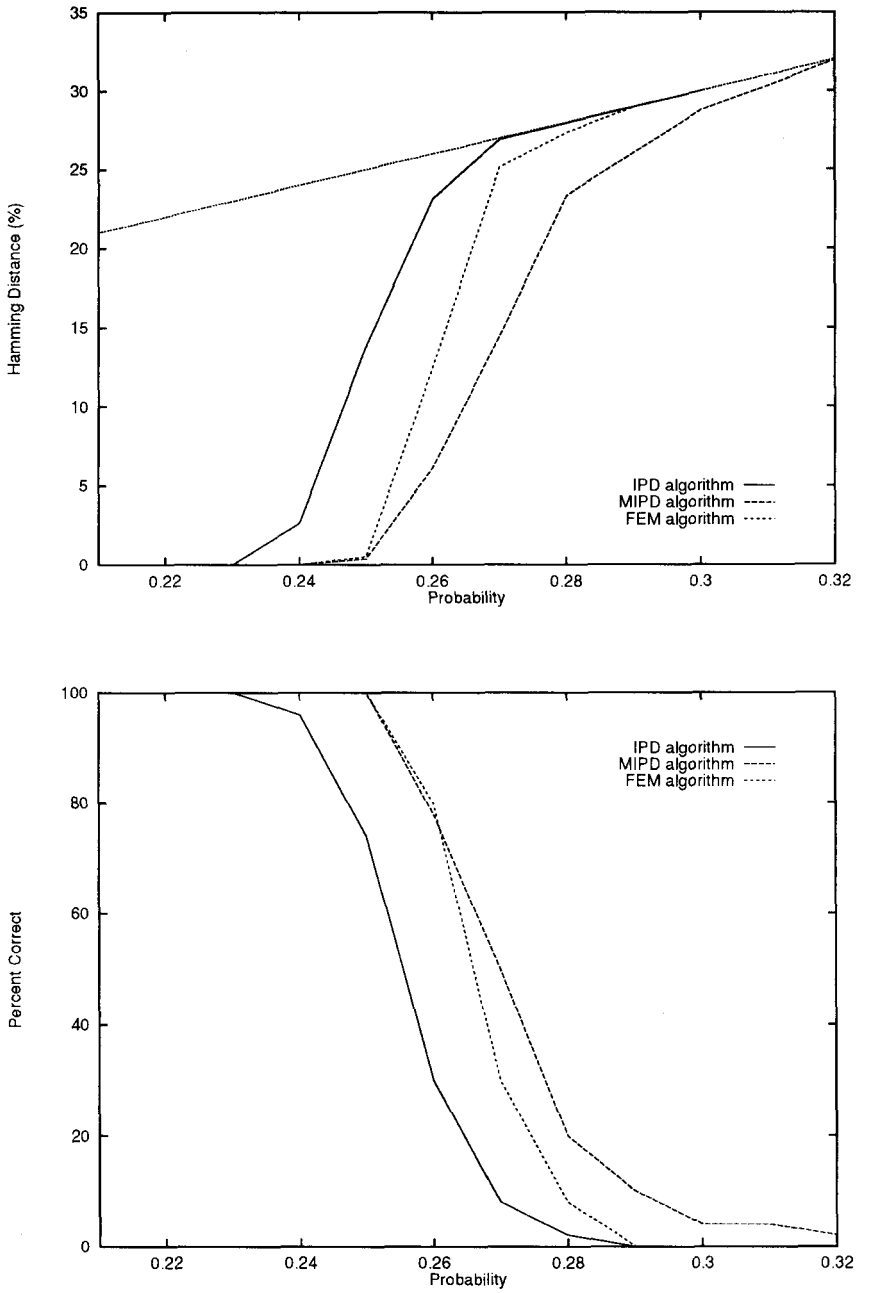

Fig. 4. Results for a shift register with 6 taps.

well with the results being consistent in all the cases except when the number of taps is six and the probability is 0.26. Here the Hamming distance result shows fast resetting to be superior but the result obtained using the information set technique shows the FEM algorithm to be (just slightly) better.

According to [11], the critical noise probability beyond which the successful iterative error-correction (particularly for the original IPD algorithm with slow resetting and orthogonal parity-checks) is not possible with a determined set of parity-checks can be approximated (the exact expression can be found in [11]) as

$$p_{\mathrm{cr}} = \frac{1 - M_w^{\frac{1}{w-1}}}{2} \tag{13}$$

where M_w is the average number of parity-checks (of weight $w+1$) per bit used in the algorithm. In the three cases examined, we have $M_2 = 22.248$, $M_4 = 33.625$ and $M_6 = 43.148$, so that p_{cr} is then given as 0.477, 0.345 and 0.265, respectively. These noise probabilities are in accordance with the experimental results shown in Figures 2-4. So, if $p > p_{\mathrm{cr}}$, then the fast correlation attacks are bound to fail on the average. However, it may be possible to extend the set of low weight parity checks by using techniques other than repeated squaring which in turn increases the critical noise probability.

5 Conclusions

In this paper two iterative error-correction techniques involving different criteria have been used to compare three fast correlation attacks. A sliding window technique which makes use of information sets is used to give useful results for probabilities exceeding the critical noise probability below which the attacks are successful. The Hamming distance is also shown to give meaningful results for this intermediate range of probabilities.

Our experimental results do not contradict the results of MacKay [6] which show that the free energy minimisation technique extends the critical noise probability (beyond which success is no longer guaranteed) from the value obtained using the standard Meier-Staffelbach approach. However, it is shown that by making a simple modification to the iterative probabilistic decoding technique such as fast resetting [4] even better results can be obtained than by FEM. From this we draw the conclusion that the fast resetting technique is superior to the FEM algorithm. Another advantage of the iterative probabilistic decoding approach is that it is easy to understand why it works since it maintains and updates a posterior probability for each bit position. Such an interpretation of MacKay's approach is not so obvious. The complexity (in terms of running time) is comparable for each of the algorithms, so that there is no considerable computational advantage of one algorithm over the other. It would be interesting to consider a combination of the two approaches in the hope of further improving the results.

Acknowledgment

The authors wish to thank David MacKay for making available the source code to his attack, without which this work would have been impossible.

References

1. V. Chepyzhov and B. Smeets, "On a fast correlation attack on stream ciphers," Advances in Cryptology - EUROCRYPT '91, *Lecture Notes in Computer Science*, vol. 547, D. W. Davies ed., Springer-Verlag, pp. 176-185, 1991.
2. G. C. Clark, Jr. and J. B. Cain, *Error-Correcting Coding for Digital Communications*. New York: Plenum Press, 1982.
3. R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.
4. J. Dj. Golić, M. Salmasizadeh, A. Clark, A. Khodkar, and E. Dawson, "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, pp. 188-202, to appear in 1996.
5. J. Dj. Golić, "On the security of shift register based keystream generators," Fast Software Encryption - Cambridge '93, *Lecture Notes in Computer Science*, vol. 809, R. J. Anderson ed., pp. 91-101, 1994.
6. D. J. C. MacKay, "A free energy minimization framework for inference problems in modulo 2 arithmetic," Fast Software Encryption - Leuven '94, *Lecture Notes in Computer Science*, vol. 1008, B. Preneel ed., Springer-Verlag, pp. 179-195, 1995.
7. J. L. Massey, *Threshold Decoding*. Cambridge, MA, MIT Press, 1963.
8. W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, pp. 159-176, 1989.
9. M. J. Mihaljević and J. Dj. Golić, "A fast iterative algorithm for a shift register initial state reconstruction given the noisy output sequence," Advances in Cryptology - AUSCRYPT '90, *Lecture Notes in Computer Science*, vol. 453, J. Seberry and J. Pieprzyk eds., Springer-Verlag, pp. 165-175, 1990.
10. M. J. Mihaljević and J. Dj. Golić, "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, pp. 527-531, 1991.
11. M. J. Mihaljević and J. Dj. Golić, "Convergence of a Bayesian iterative error-correction procedure on a noisy shift register sequence," Advances in Cryptology - EUROCRYPT '92, *Lecture Notes in Computer Science*, vol. 658, R. A. Rueppel ed., Springer-Verlag, pp. 124-137, 1993.
12. T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Comput.*, vol. 34, pp. 81-85, Jan. 1985.
13. K. Zeng and M. Huang, "On the linear syndrome method in cryptanalysis," Advances in Cryptology - CRYPTO '88, *Lecture Notes in Computer Science*, vol. 403, S. Goldwasser ed., Springer-Verlag, pp. 469-478, 1990.
14. M. V. Živković, "On two probabilistic decoding algorithms for binary linear codes," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1707-1716, Nov. 1991.