Fast Low Order Approximation of Cryptographic Functions

Jovan Dj. Golić *

Information Security Research Centre, Queensland University of Technology
GPO Box 2434, Brisbane Q 4001, Australia
School of Electrical Engineering, University of Belgrade
Email: golic@fit.qut.edu.au

Abstract. A fast and efficient procedure for finding low order approximations to large boolean functions, if such approximations exist, is developed. The procedure uses iterative error-correction algorithms for fast correlation attacks on certain stream ciphers and is based on representing low order boolean functions by appropriate linear recurring sequences generated by binary filter generators. Applications and significance of the proposed method in the analysis and design of block and stream ciphers are also discussed.

1 Introduction

Encryption and decryption functions of binary block and stream ciphers in their various modes of operation are necessarily based on boolean functions. The boolean functions are secret key dependent in block ciphers and self-synchronizing stream ciphers and may be key independent in synchronous stream ciphers where the initial state is key dependent. Linear approximations of boolean functions in stream cipher applications have been studied in [22], [23], [21], [26], [16], [2] for memoryless combiners, in [24], [21], [2] for nonlinear filter generators, in [17], [6], [7] for combiners with memory, and in [7] for arbitrary keystream generators. These results are related to correlation attacks [23] or fast correlation attacks [15], [27] on shift register based keystream generators. On the other hand, a number of authors have investigated linear approximations of boolean and vectorial boolean functions in product block ciphers, and a breakthrough in this area was made by Matsui in his seminal work [13], initiating many follow-up papers on similar problems. Matsui developed an iterative (dynamic programming) method for finding linear approximations to encryption functions in product block ciphers and was the first to realize that even linear approximations with very small correlation coefficients can be used to reduce the uncertainty of the secret key. Maurer [14] has analyzed the security of self-synchronizing stream ciphers against the chosen-ciphertext attack and pointed out the importance of linear and other low order approximations to the key dependent feedback

* 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.

U. Maurer (Ed.): Advances in Cryptology - EUROCRYPT '96, LNCS 1070, pp. 268-282, 1996.

function for approximate reconstruction of unknown plaintext from a given ciphertext which in turn may lead to the complete plaintext recovery by using the plaintext redundancy. The reason for this is that any boolean function of low algebraic/nonlinear order can be described and evaulated in terms of a relatively low number of coefficients in its algebraic normal form (ANF) even if the number of input variables is large. Of course, approximate decryption by means of low order approximations to decryption functions may also be possible for any block cipher as well [20].

The task of finding low order approximations to boolean functions necessarily involves computing or estimating the correlation coefficient between a given boolean function and a candidate boolean function of low algebraic order, where the correlation coefficient between any two boolean functions $f(X)$ and $g(X)$ of n variables $X = (x_1, \ldots, x_n)$ is as usual (e.g., see [21], [16]) defined as

$$c(f,g) \;=\; 2^{-n} \sum_{X} (-1)^{f(X)}(-1)^{g(X)} \;=\; \Pr\{f = g\} - \Pr\{f \neq g\}. \qquad (1)$$

The functions f and g are not correlated if $c(f,g) = 0$, and are strongly correlated if $c(f,g)$ is large in magnitude (i.e., close to ± 1). The objective of low order approximation of a given boolean function f is to find a boolean function g of low algebraic order such that the correlation coefficient $c(f,g)$ is relatively large. To compute $c(f,g)$ exactly for a candidate g, one has to examine all 2^n possible arguments, but to estimate it reliably only $O(1/c(f,g)^2)$ argument values are needed, under a reasonable probabilistic assumption that 2^n values of $g(X)$ are chosen independently at random so that $\Pr\{f = g\} = (1 + c(f,g))/2$. Accordingly, the correlation coefficient is considered to be large in magnitude if $|c(f,g)|$ is much bigger than $2^{-n/2}$. The exhaustive search over all $2^{\sum_{i=0}^{r} \binom{n}{i}}$ possible candidate functions g of order r or smaller is not feasible for moderately large n (e.g., $n \geq 64$) which is the case in cryptographic applications. It should be noted that for linear/affine approximations ($r = 1$) one may apply the fast Walsh transform algorithm for exact computation of the correlation coefficients between f and all the linear functions, with the computational complexity $O(n2^n)$ instead of 2^{2n} (e.g., see [21]). A similar algorithm for an arbitrary order r is not known. In any case, for 2^n very large the problem is how to find good candidate functions, if they exist. The problem is difficult and may even seem impossible to solve.

Maurer [14] pointed out that boolean functions of n variables and of order at most r can be regarded as codewords of a binary rth-order Reed-Muller code with parameters $(2^n, \sum_{i=0}^{r} \binom{n}{i}, 2^{n-r})$, see [25], where the information bits are the coefficients in the ANF of these functions. The considered problem can then be regarded as one of decoding such linear block codes, i.e., the minimum distance decoding of such codes is equivalent to finding a low order approximation to a given boolean function with the maximum correlation coefficient. Since the minimum distance is 2^{n-r}, one may correct any $2^{n-r-1} - 1$ or less errors by such codes which means that it is theoretically possible to determine uniquely the best rth-order approximation g to a given boolean function f if the correlation coefficient $c(f,g)$ is bigger than $1 - 2^{-r}$. However, as noted in [14], standard

decoding procedures [25] for Reed-Muller codes (majority-logic decoding based on appropriate orthogonal parity-checks) are not feasible for large 2^n. Another important distinction is that in cryptographic applications the maximum magnitude of the correlation coefficients to rth-order boolean functions is expected to be much smaller than $1 - 2^{-r}$. Such approximations may no longer be useful for approximate decryption, but may be used in cryptanalytic attacks on the secret key. For example, Matsui [13] has shown that linear approximations to encryption functions in block ciphers with the correlation coefficients even close to $2^{-n/2}$ can be used to reduce the uncertainty of the secret key. Similar conclusions may hold for arbitrary low order approximations and for self-synchronizing and synchronous stream ciphers as well.

Maurer [14] has proposed two local decoding algorithms for Reed-Muller codes, one for affine approximations ($r = 1$) and the other for approximations of an arbitrary order r. Both are feasible for large 2^n and essentially consist in majority-logic decoding based on appropriate subsets of orthogonal parity-checks for the ANF coefficients of a boolean function g of order at most r (the information bits). The first algorithm is based on the parity-checks involving three bits only, whereas the second one is based on decoding short Reed-Muller codes derived from the long one and then on making a majority-logic decision over all the short codes for every coefficient of g. The short codes are obtained by setting some of the input variables to zero. Maurer suggested that with high probability the first algorithm can work even if the correlation coefficient is much smaller than $1/2$ and that the second one may find the best rth-order approximation with the correlation coefficient bigger than $1 - 2^{-r}$.

Another, information set decoding (e.g., see [3]) approach to the low order approximation problem is taken in [20]. For approximations of order at most r, the information sets suggested are Hamming spheres of radius r. For an approximation with the correlation coefficient c, the expected number of sphere samples to be examined to find with high probability an error-free one can be estimated as $(2/(1+c))^{\sum_{i=0}^{r}\binom{n}{i}}$, which is feasible only for relatively large c.

The main objective of this paper is to develop a fast procedure for finding low order approximations to large boolean functions that can be successful even if the correlation coefficient of the best approximation is very small (e.g., much smaller than $1 - 2^{-r}$ for an rth-order approximation). Our solution is based on appropriate parity-checks, but unlike [14], the decision (majority-logic rule or maximum posterior probability rule) is not made on the ANF coefficients of the approximation function (information bits), but on all the codeword bits involved in the parity-checks. This implies that the observed set of the codeword bits should be closed with respect to all the parity-checks used. The decision process is then repeated iteratively, each time recomputing all the parity-checks, which greatly increases the error-correction capability, i.e., enables one to successfully deal with much smaller correlation coefficients of the best approximation. To the same end, it is also important that the parity-checks be chosen with as few terms as possible. After the iterative algorithm is completed, the number of remaining errors, i.e., the Hamming distance to the best low order approximation

is considerably reduced. This then enables us to apply a simple information set decoding technique to reconstruct the codeword corresponding to the best approximation, if the correlation coefficient is not too small to be detected. Finally, the coefficients in its ANF are then obtained by an appropriate linear transform.

It remains to explain how to construct the codewords of the underlying linear code and how to find the required parity-checks. For a boolean function f of n variables to be analyzed, take a linear feedback shift register (LFSR) of length $m \geq n$ with a primitive feedback polynomial and a fixed initial state and select any consecutive n LFSR stages to form a filter generator. Then for each boolean function g of n variables and of order at most r with the zero constant term in its ANF, form a filter generator with g as a filter function and produce a sequence of length N. In view of an old result from [10], it follows that every such sequence satisfies a linear recurrence determined by a binary polynomial h_r of degree $m_r = \sum_{i=1}^{r} \binom{m}{i}$, provided that $N > m_r$. The sequences are different and constitute a linear code with $k_r = \sum_{i=1}^{r} \binom{n}{i}$ information bits (i.e., the coefficients in the ANF of g). The parity-checks are then constructed from low weight polynomial multiples of h_r where the weight of a binary polynomial is defined as the number of its nonzero terms. Note that m greater than n can make finding the low weight and low degree polynomial multiples of h_r easier (preferably, h_r itself should have a low weight).

Then, produce a sequence of length N by the filter generator with the given boolean function f as a filter function. This sequence is regarded as a received codeword at the output of a binary symmetric channel with the noise probability $(1 - c)/2$ corresponding to the assumed correlation coefficient c. Since c is assumed to be positive, we have to run the decoding procedure twice: once for the received codeword and second time for its binary complement. The iterative probabilistic or majority-logic decoding algorithm with information set decoding then yields a linear recurring sequence satisfying h_r that corresponds to an approximation to f of order at most r with the correlation coefficient close to c or bigger, if it exists. A condition for the minimum c that can be detected is pointed out. The solution need not be unique, especially if $r > 1$. Interestingly, it thus turns out that the low order approximation problem for boolean functions can essentially be reduced to the problem of fast correlation attacks on LFSR's with appropriate feedback polynomials. For $r = 1$ the feedback polynomials are primitive, but for $r > 1$ they are not. The basic iterative probabilistic decoding algorithm for fast correlation attacks was introduced in [15], while a similar iterative majority-logic decoding algorithm was independently proposed in [27]. Both the algorithms have origins in iterative error-correction decoding procedures given in [5], [11], and [3]. In a number of follow-up papers, these algorithms have been modified and further analyzed, e.g., see [4], [1], [18], [19], [28], [8]. The important difference that greatly facilitates the success of the low order approximation procedure is that the feedback polynomial, h_r, can be chosen so as to maximize the number of parity-checks associated with parity-check polynomials of low weight and not too large a degree, and finding such parity-checks can be done in precomputation time.

The rest of the paper is organized as follows. A more detailed description of the background work from [14] and [20] is presented in Section 2. The representation of low order boolean functions by filter generators is explained in Section 3, whereas the relation between low order approximation of boolean functions and fast correlation attacks is investigated in Section 4. Significance and potential applications of the proposed method in the design and analysis of block and stream ciphers are discussed in Section 5.

2 Background Work

Maurer [14] has proposed two local decoding algorithms for Reed-Muller codes that are feasible for large 2^n, one for affine approximations ($r = 1$) and the other for approximations of an arbitrary order r. Both essentially consist in majority-logic decoding based on appropriate subsets of orthogonal parity-checks for the ANF coefficients of a boolean function of order at most r (the information bits). Regarding the affine approximations, note that every $(2^n, n+1, 2^{n-1})$ first-order Reed-Muller code can be reduced to a $(2^n - 1, n, 2^{n-1})$ code being the dual of a binary Hamming code, by deleting the information/codeword bit corresponding to the constant term of an affine function. This means that linear boolean functions correspond to codewords of the dual of a binary Hamming code, see [25]. Majority-logic decoding of such codes is based on 2^{n-1} orthogonal parity-checks for each information bit, where each parity-check involves three bits only. The algorithm proposed in [14] for decoding long first-order Reed-Muller codes is then the same except that it uses subsets of the parity-checks, for the decoding to be feasible for large 2^n. The best linear/affine approximation with the correlation coefficient bigger than $1/2$ is guaranteed to be found by this algorithm if all the parity-checks are used. Maurer suggested that this is with high probability true even if the correlation coefficient is much smaller than $1/2$.

However, for an arbitrary r, the choice of a small subset of the parity-checks to be used in majority-logic decoding is not as simple as for $r = 1$. An interesting solution given in [14] is based on decoding short Reed-Muller codes derived from the long one and then on making a majority-logic decision over all the short codes for every information bit. The short codes of length 2^l are obtained by setting $n - l$ input variables to zero. It is recommended in [14] to choose sufficiently many subsets of l variables to cover every information bit of the long code if l is large, and to use all $\binom{n}{l}$ of them if l is relatively small. It is suggested that such an algorithm may with high probability find the best rth-order approximation with the correlation coefficient bigger than $1 - 2^{-r}$. As opposed to the case $r = 1$, Maurer did not discuss the situation when the correlation coefficient is smaller than $1 - 2^{-r}$. While it is intuitively clear that the proposed algorithm can work, the conditions for success are not quite clear (e.g., the choice of l).

Another, information set decoding (e.g., see [3]) approach to the low order approximation problem is suggested in [20]. It is well-known [25] that the ANF coefficients of a boolean function of order at most r can be uniquely determined

from its values in the Hamming sphere of radius r around the all-zero vector by a linear transform. It is observed in [20] that the same holds for the Hamming sphere of radius r around any input vector as a center, where the corresponding linear transform incorporates the boolean derivatives of the function with respect to the center vector. In coding terms, the sets of codeword bits corresponding to these Hamming spheres represent particular examples of the information sets (there are many others), and the main point of information set decoding is to look for error-free information sets. Each Hamming sphere thus gives a candidate low order approximation to a given boolean function, and for each candidate the correlation coefficient is estimated on a set of $O(1/c^2)$ arguments, for an assumed value c. A candidate function with the maximum magnitude of the correlation coefficient estimate is then picked as a low order approximation. If an approximation with the correlation coefficient c exists, then the expected number of sphere samples to be examined to find with high probability an error-free one can be approximated as $(2/(1+c))\sum_{i=0}^{r}\binom{n}{i}$. The computational complexity is thus prohibitively high, so that the method may work only for very large c, given that n is moderately large and r is not very small.

3 Low Order Functions and Filter Generators

In this section, it is defined how to construct the codewords of the underlying linear code used for low order approximation and how to find the desired parity-checks. For a boolean function of n variables to be analyzed, take a linear feedback shift register (LFSR) of length $m \geq n$ with a primitive feedback polynomial h. Choose and fix an arbitrary nonzero initial state and select arbitrary consecutive n out of m LFSR stages to form a filter generator. Recall [21] that a binary filter generator is a keystream generator consisting of a single binary LFSR and a boolean function whose inputs are taken from some shift register stages to produce the output. Such a generator realizes a linear/nonlinear feedforward transform of an LFSR sequence. More precisely, let $x = (x(t))_{t=-m}^{\infty}$ be a binary maximum-length sequence of period $2^m - 1$ generated from the assumed initial state $(x(t))_{t=-m}^{-1}$, let $f(x_1, \ldots, x_n)$ be a boolean function of n, $n \leq m$, input variables, and let the first n LFSR stages define the inputs to the filter function. Then the output sequence $y = (y(t))_{t=0}^{\infty}$ of the corresponding filter generator is defined by $y(t) = f(x(t-1), \ldots, x(t-n))$, $t \geq 0$.

For any $r \geq 1$ (for low order approximations r/n should be small), define a collection $\mathcal{G}_r$ of 2^{k_r}, $k_r = \sum_{i=1}^{r}\binom{n}{i}$, filter generators with the same initial state by using all possible boolean functions g of n variables and of order at most r, where the constant term in the ANF of g is assumed to be equal to zero. For simplicity, the parameters n and m are dropped from the notation. Recall that the algebraic normal form (ANF) of g is defined as $g(x_1, \ldots, x_n) = \sum_S a_S \prod_{i \in S} x_i$ where the sum is over all index sets $S \subseteq \{1, \ldots, n\}$ of cardinality at most r, and the coefficients a_S are binary. The output sequences produced by $\mathcal{G}_r$ are periodic with a period equal to $2^m - 1$ or to a factor of $2^m - 1$ and are all different since g are different and all 2^n possible arguments of g are used to produce the output

bits. Furthermore, the set, $\mathcal{S}_r$, of 2^{k_r} different output sequences produced by $\mathcal{G}_r$ is a binary vector space of dimension k_r, because the ANF uniquely represents g and every g can be expressed as a sum of product boolean functions, whose ANF contains a single product term. The corresponding generating sequences for $\mathcal{S}_r$ are the k_r sequences produced by all k_r possible product boolean functions of n variables and of order at most r. This means that the output sequence produced by any g can be expressed as a linear combination of these generating sequences where the coefficients are exactly the ANF coefficients of g (see [21]).

Moreover, due to a well-known result from [10], the sequences from $\mathcal{S}_r$ all satisfy a linear recurrence defined by an appropriate binary polynomial h_r of degree $m_r = \sum_{i=1}^{r} \binom{m}{i}$. More precisely, if α is a root of h in a splitting field $\mathrm{GF}(2^m)$, then the roots of h_r in $\mathrm{GF}(2^m)$ are exactly all the powers α^e where e is an integer with an m-bit long binary expansion that contains at least one and at most r ones. Accordingly, h_r is the least common multiple of the minimum binary polynomials of all such powers. It can be obtained by standard algebraic techniques or, simply, by using the Berlekamp-Massey algorithm [12] over the binary field. Recall that the Berlekamp-Massey algorithm yields the shortest LFSR that generates a given finite field sequence of a finite length (its feedback polynomial is called the minimum feedback polynomial of the sequence, and is unique if its length is at most one half of the sequence length). It is shown in [21] that the minimum feedback polynomial of every sequence in $\mathcal{S}_r$ produced by any product boolean function g of i adjacent variables, $1 \leq i \leq r$, necessarily contains as roots all the powers α^e such that the binary expansion of e contains exactly i ones. For each $1 \leq i \leq r$, one may then take exactly one product boolean function of the first i variables, and h_r is then the least common multiple of the minimum feedback polynomials of the corresponding r binary sequences produced by these r product functions. To determine the individual feedback polynomials uniquely, it suffices to take the first $2m_r$ bits of these sequences and to apply the Berlekamp-Massey algorithm. The least common multiple can then be found efficiently by the Euclidean division algorithm. Alternatively, one may as well apply the multisequence Berlekamp-Massey algorithm [2] to these r product sequences of length $2m_r$ and thus directly obtain h_r.

Consequently, $\mathcal{S}_r$ is a set of 2^{k_r} binary sequences which can be generated by an LFSR of length m_r and the feedback polynomial h_r from 2^{k_r} different initial states. Each sequence in $\mathcal{S}_r$ is then uniquely determined by any m_r consecutive bits, by the forward and backward linear recurrences. Consider now a set of truncated output sequences of length $N > m_r$ produced by $\mathcal{G}_r$ by taking the first N output bits only. Since the sequences are different, this set is clearly a binary (N, k_r) linear code, denoted as $\mathcal{C}_r^N$, whose dimension is exactly k_r. The code $\mathcal{C}_r^N$ can be regarded as a subcode of a truncated cyclic code with the parity-check polynomial h_r. This cyclic code is a generalization of a cyclic representation of $(2^m - 1, \sum_{i=1}^{r} \binom{m}{i}, 2^{m-r})$ Reed-Muller codes associated with boolean functions of m variables with the zero constant term in the ANF and with the codeword bit corresponding to the all-zero vector omitted, see [9]. If $m = n$, then $m_r = k_r$ and $\mathcal{C}_r^N$ is a truncated cyclic code itself. The desired parity-checks correspond to low weight polynomial multiples of h_r.

It is allowed that m can be greater than n in order to make easier finding the low weight polynomial multiples of h_r whose degree is relatively small or moderately large, so that the length N need not be too large. For example, for $r = 1$ the parity-check polynomial h_r coincides with the LFSR polynomial h. It is therefore preferable that h be a trinomial, which may not exist for $m = n$ but exists for $m > n$. Of course, obtaining trinomial or low weight primitive polynomials of degree m by exhaustive search is relatively easy if the factorization of $2^m - 1$ is known. One may develop special methods for finding low weight polynomials h_r for any $r \geq 1$ as well. Starting from any determined low weight polynomial multiple of h_r, other low weight polynomial multiples can then be produced by the squaring technique proposed in [15]. In general, finding low weight polynomial multiples of h_r of not too large a degree can be performed by systematic search based on computing the residues of the single term (power) polynomials modulo h_r. For higher degrees, one may use the meet-in-the-meedle technique [15] or algorithms for computing discrete logarithms in fields of characteristic two, as suggested in [27]. In any case, finding an appropriate (optimal) primitive polynomial h and low weight polynomial multiples of not too large a degree for the corresponding polynomial h_r can be done in precomputation time. As usual [15], [1], any determined low weight polynomial multiple of weight W actually defines a set of W parity-checks corresponding to its different phase shifts. The parity-checks should be tested for orthogonality in which case some of them may be discarded. If N is large enough, the same set of the parity-checks (except for a phase shift) is used for most the codeword bits, and near the ends the set is reduced appropriately.

As noted above, since the dimension of the code $\mathcal{C}_r^N$ is k_r if $N > m_r$, every codeword in $\mathcal{C}_r^N$ uniquely determines a sequence in $\mathcal{S}_r$ and, hence, a boolean function g of order at most r as well. In turn, every codeword is uniquely determined by any m_r consecutive bits or by any other information set. Therefore, the ANF coefficients of g can be obtained from any m_r consecutive bits in the corresponding codeword. More precisely, each out of k_r ANF coefficients of g can be expressed as a linear function of any m_r consecutive bits from the associated codeword. To find these linear functions, it suffices to take the first m_r codeword bits (as variables) and to form a system of linear equations in the unknown ANF coefficients by expressing the sequence of m_r codeword bits as a linear combination, with unknown coefficients, of the k_r sequences of length m_r produced by all k_r possible product boolean functions of n variables and of order at most r. The corresponding system of m_r linear equations in k_r unknowns has rank k_r and, hence, has a unique solution for the unknown ANF coefficients given the first m_r codeword bits. The system is sparse and can be easily solved by standard techniques even if k_r and m_r are very large. This is executed in the precomputation time and as a result one thus obtains and stores a binary $k_r \times m_r$ matrix $\mathbf{A}_r$ defining a linear transform for the unknown ANF coefficients of g in terms of the first m_r bits of a given codeword in $\mathcal{C}_r^N$.

4 Low Order Approximations and Fast Correlation Attacks

In the previous section, a filter generator based representation of low order boolean functions as codewords in an appropriate binary linear code of essentially cyclic structure was defined. In this section, this representation is used to propose a solution to the low order approximation problem for large boolean functions in terms of iterative probabilistic or majority-logic decoding algorithms used in fast correlation attacks on certain stream ciphers. Assume that a boolean function f of n variables is given where n may be large. More precisely, we assume that f can be evaluated on a desired set of arguments of cardinality much smaller than 2^n.

Accordingly, as described in the previous section, choose an LFSR with an appropriate primitive feedback polynomial h of degree $m \geq n$, select and fix the LFSR initial state, and then, in terms of filter generators, define a binary (N, k_r) linear code $\mathcal{C}_r^N$, where $k_r = \sum_{i=1}^{r} \binom{n}{i}$ and $N > m_r$, $m_r = \sum_{i=1}^{r} \binom{m}{i}$. Its codewords satisfy a binary polynomial h_r of degree m_r and uniquely represent boolean functions g of n variables and of order at most r. Then compute a binary $k_r \times m_r$ matrix $\mathbf{A}_r$ defining a linear transform for the unknown ANF coefficients of g in terms of the first m_r bits of a given codeword in $\mathcal{C}_r^N$. The main precomputational effort is to determine a set, Λ_r, of low weight parity-check polynomials whose degrees should be as small as possible. Given n, a degree m and a primitive polynomial h should be optimized accordingly.

Then form a filter generator from the same LFSR by using the given function f as a filter function and produce a sequence of length N. This sequence is regarded as a received codeword for $\mathcal{C}_r^N$ at the output of a binary symmetric channel with the noise probability $p = (1 - c)/2$, corresponding to the assumed correlation coefficient c. So, the crucial point of our approach is to observe the values of f for the argument values corresponding to successive states of an LFSR with an appropriate primitive feedback polynomial. The objective of low order approximation of f is to find a boolean function g of order at most r such that the absolute value of the correlation coefficient $c(f, g)$ is c or larger, if such g exists. In coding terms, this problem then reduces to the minimum distance decoding of $\mathcal{C}_r^N$, i.e., to finding a codeword at the minimum Hamming distance (or close) to the received codeword. Since c is assumed to be positive, we have to run the decoding procedure twice: once for the received codeword and second time for its binary complement. Alternatively, one may incorporate the polynomial $1 + x$ as a factor of h_r to include the boolean functions g with the nonzero constant term in the ANF (then k_r is increased by one). As the codewords of $\mathcal{C}_r^N$ satisfy the linear recurrence defined by h_r, the minimum distance decoding of this code is then essentially similar to the problem of the LFSR initial state reconstruction from a noisy LFSR sequence which arises in correlation attacks [23] on certain stream ciphers based on memoryless combiners. In this case, the LFSR feedback polynomial is h_r, which is primitive only if $r = 1$. Since the number, 2^{k_r}, of possible LFSR initial states (i.e., the number of low order approximations g) is

very large, only fast decoding procedures are of interest, which is essentially the situation dealt with in fast correlation attacks.

The solution is in iterative probabilistic or majority-logic decoding procedures with information set decoding. The basic iterative probabilistic decoding algorithm for fast correlation attacks was introduced in [15], while a similar iterative majority-logic decoding algorithm was independently proposed in [27]. Both the algorithms have origins in iterative error-correction decoding procedures given in [5], [11], and [3]. In a number of follow-up papers, these algorithms have been modified and further analyzed, e.g., see [4], [1], [18], [19], [28], [8]. The unknown codeword is assumed to have the form $\mathbf{z} + \mathbf{e}$, where $\mathbf{z} = (z_i)_{i=1}^N$ is the received codeword produced by f and $\mathbf{e} = (e_i)_{i=1}^N$ is the unknown error vector to be reconstructed. Iterative error-correction algorithms consist in making iterative decisions on all the bits in $\mathbf{e}$ based on the parity-check values being updated after every decision on $\mathbf{e}$. In majority-logic error-correction, the majority-logic decision rule is applied based on the number of satisfied (zero valued) parity-checks. The assumed value of the correlation coefficient c is not used, and the algorithm works well if c is not too small and, especially, if the parity-checks have the same weight. For smaller values of c and parity-checks of different weights, one should apply probabilistic error-correction, where the statistically optimal, maximum posterior probability decision rule is employed for each bit of $\mathbf{e}$ which minimizes the symbol error-rate in the first iteration step. A probabilistic model is used in which $\mathbf{e}$ is assumed to be a sequence of independent binary random variables (bits) in each iteration step. Initially, the error bits are assumed to be identically distributed with $\Pr\{e_i = 1\} = p = (1 - c)/2$. Let in each iteration step, p_i and $\hat{p}_i$ denote the prior and posterior probabilities that $e_i = 1$, respectively, and let the corresponding correlation coefficient be defined as $c_i = 1 - 2p_i$. Let $A_{r,i}$ denote the set of the parity-checks used for e_i and let $s(\lambda)$ denote the binary value of a parity-check $\lambda \in A_{r,i}$. Let $c(\lambda)$ be the correlation coefficient associated with $\lambda \in A_{r,i}$ which is defined as the product of the correlation coefficients c_j for all the error bits e_j, $j \neq i$, involved in λ (since the error bits are assumed to be independent, $c(\lambda)$ is exactly the correlation coefficient of their modulo 2 sum [5]). If the parity-checks in $A_{r,i}$ are orthogonal, then their random values are independent when conditioned on the value of e_i, so that the posterior probability $\hat{p}_i$ is then for each $1 \leq i \leq N$ determined by

$$\frac{\hat{p}_i}{1 - \hat{p}_i} = \frac{p_i}{1 - p_i} \prod_{\lambda \in A_{r,i}} \left(\frac{1 - c(\lambda)}{1 + c(\lambda)} \right)^{1 - 2s(\lambda)} . \tag{2}$$

The same expression may be used for nonorthogonal parity-checks as well.

The main point of the iteration process is then to use the posterior probabilities computed in the current step as the prior probabilities for the next step. Equivalently (for orthogonal parity-checks [28], [18]), one may perform error-correction in each iteration step by using the maximum posterior probability decision rule: if $\hat{p}_i > 0.5$, then e_i is set to 1, the observed bit z_i is complemented, and $\hat{p}_i$ is set to $1 - \hat{p}_i$. With error-correction in each step, the experiments [15], [18], [19], [8] show that the posterior probabilities relatively quickly converge to

the zero value for most positions i on a codeword length. This fact is a consequence of the fixed points of the mapping defined by (2) and does not necessarily mean that all or most the errors are thus corrected. But, hopefully, the number of errors (i.e., the Hamming distance to the best low order approximation) is reduced. One may then reset all the error probabilities to the initial value p [15] and repeat the iteration process for several times until all or most the errors are corrected. Experiments [8] show that some improvement is achieved with a fast resetting algorithm where resetting is performed according to the cumulative number of error-corrections, before the convergence point is reached. Alternatively, one may also use a modified equation (2) so that the first product term is always equal to $p/(1-p)$ or one, see [5], [28].

Let c_* denote the correlation coefficient $c(f,g)$ for the best low order approximation g to f. Experimental evidence reveals that most the errors are corrected if c_* is greater than an upper threshold value and if the assumed c is smaller than and relatively close to c_*. On the other hand, if c_* is smaller than or equal to a lower threshold value, then the number of errors can not be reduced, which means that the best low order approximation can not be found by using the assumed set of parity-checks. Between the two thresholds, the number of errors may be reduced, but remains relatively high on the average. According to a characterization of the lower threshold value obtained in [19], a necessary condition for a successful iterative error-correction can for relatively small c_* be approximated as

$$\sum_w \overline{M}_w \, c_*^{w-1} \; > \; 1 \tag{3}$$

where $\overline{M}_w$ denotes the average number of the parity-checks of weight $w+1$ in $A_{r,i}$ over all positions $1 \leq i \leq N$. Experiments show that this condition is close to being sufficient as well. If c_* is smaller than the threshold implicit in (3), then the number of parity-checks must be increased. The significant gain obtained by iterative error-correction as compared with single-step error-correction can be seen in the case where all the parity-checks have the same weight (e.g., $w=2$, see [14] for $r=1$). Then the condition (3) becomes $\overline{M}_w c_*^{w-1} > 1$, whereas for single-step error-correction for any m_r consecutive codeword bits the condition for success can be expressed as $m_r \overline{M}_w c_*^{2w} > 1$. So, by iterative error-correction algorithms one may find low order approximations with very small correlation coefficients, much smaller than $1-2^{-r}$, if they exist.

After a number of iteration rounds, the iterative error-correction is completed, and an estimate $\hat{\mathbf{z}}$ of the closest codeword is obtained. In general, it may contain a number of residual errors, which are then corrected by the information set decoding procedure. One may use simple information sets of m_r consecutive bits in $\hat{\mathbf{z}}$ ($N - m_r + 1$ of them) in search of an error-free one. For each assumed information set one then computes the remaining codeword bits by the forward and backward linear recurrence, and then the correlation coefficient between each such candidate codeword and the received one is estimated by using $O(1/c^2)$ codeword bits. Unlike [20], this can be done without computing the ANF coefficients of a low order boolean function g corresponding to the

assumed information set. A candidate codeword is accepted if the computed correlation coefficient estimate is consistent with the assumed value c or bigger. If an accepted codeword is found, then the ANF coefficients of the corresponding low order approximation g are computed from the first m_r codeword bits by the matrix $\mathbf{A}_r$.

We have thus shown that the problem of finding a low order approximation to a large boolean function f can be solved by iterative probabilistic decoding algorithms combined with information set decoding both applied to a sequence of values of f for the argument values taken from successive states of an LFSR with an appropriate primitive feedback polynomial, h. However, there is a number of important differences from standard fast correlation attacks. The main advantage is that the feedback polynomial, h_r, of the underlying LFSR can be chosen in precomputation time so as to optimize the success of iterative probabilistic decoding. If $r = 1$, then $h_r = h$ is a primitive polynomial and the situation is very much similar to the one in fast correlation attacks on memoryless combiners. More precisely, this is the case if the best affine approximation to f is unique and if its correlation coefficient, c_*, to f is 'considerably' bigger than for other affine functions. Since the number, n, of input variables of f is large, this is very likely to be true if c_* is much bigger than $2^{-n/2}$. If there exists a number of affine functions with mutually close correlation coefficients to f much bigger than $2^{-n/2}$ (the case of multiple affine/linear approximations), then the situation is more similar to the one in fast correlation attacks on nonlinear filter generators, see [4]. The problem in this case is that different affine functions are mutually uncorrelated, which may confuse the iterative error-correction algorithms. On the other hand, unlike [4], [24], we are here satisfied with any found affine approximation, not all of them. The problem may be resolved by using more complicated information sets, e.g., chosen from a number of the most significant probabilities (the smallest ones after the complementation) in the error probability vector after one or just a few iteration steps.

If $r > 1$, then h_r is not a primitive polynomial and its degree m_r is large if n is large. Fortunately, the influence of large m_r on the success of iterative error-correction is insignificant, see (3). Only the final, information set decoding stage is slightly affected. Also, the precomputational effort to find the low weight parity-checks and to compute the matrix $\mathbf{A}_r$ is increased. In addition, for $r > 1$, due to the minimum distance property of rth-order boolean functions, it should be expected that the best rth-order approximation to f with a correlation coefficient much bigger than $2^{-n/2}$, if it exists, is essentially not unique. Namely, for any rth-order function with the correlation coefficient c_* to f, it is very likely that there exist a number of rth-order functions with the correlation coefficients to f of the order of $(1 - 2^{-(r-1)})\, c_*$, which may be considered 'close' to c_*. However, due to the same effect, these multiple rth-order approximations are mutually correlated with the correlation coefficients close to $1 - 2^{-(r-1)}$, which is less confusing for the iterative error-correction algorithms.

There is another potential advantage to be used for any $r \geq 1$. For any chosen primitive polynomial h, one may use different LFSR initial states to form a set of

filter generators giving rise to a set of the corresponding codes to which iterative probabilistic decoding algorithms can be applied simultaneously. Moreover, one may also deal with different polynomials h. Suppose that none of the codeword estimates has yielded the desired low order approximation. It is now possible to combine individual codeword estimates by an appropriate procedure, e.g., by a majority-logic decision on the corresponding ANF coefficients obtained by applying the matrix $\mathbf{A}_r$ to the first m_r bits of each of the codeword estimates. This can be useful especially in the case of multiple low order approximations.

5 Conclusions

A solution to a difficult, low order approximation problem for large boolean functions is developed by establishing a connection between binary filter generators and iterative error-correction algorithms used in fast correlation attacks on certain stream ciphers. A boolean function f to be approximated has to be evaluated on a set of arguments corresponding to successive states of an LFSR with an appropriate primitive feedback polynomial. The proposed procedure is fast and can with high probability find low order approximations with very small correlation coefficients to f, if such approximations exist. The correlation coefficient can be much smaller than $1 - 2^{-r}$, which is the lower bound for the best rth-order approximation to be unique necessarily. Its magnitude is practically limited only by the computational power available.

The proposed method may have wide cryptographic applications. First, if f is a secret key dependent decryption function, then in the chosen-ciphertext scenario, the method can be used for approximate decryption of self-synchronizing stream ciphers [14] or block ciphers [20], provided the correlation coefficient of the found low order approximation is big enough with respect to the plaintext redundancy. More importantly, the method can generally be used for the analysis of any large boolean functions in various cryptographic applications. In this case, f is known and can be evaluated on an arbitrary set of arguments. In block ciphers, f can be an encryption boolean function of both the secret key and plaintext input variables. Low order approximations even with very small correlation coefficients may then be used for the secret key reconstruction from a sufficient number of known (possibly chosen) plaintext-ciphertext pairs, as is demonstrated in [13] for linear approximations. Alternatively, apart from the black-box approach, in a structure-based approach, one may find low order approximations to large boolean functions employed in one-round functions of product block ciphers and then combine them to obtain an overall approximation for the whole cipher, see [13] for $r = 1$. Of course, rth-order boolean functions are not closed under composition for $r > 1$, but any resulting approximation, of not necessarily low order, may be a basis for the secret key reconstruction as long as the number of nonzero terms in its ANF is not too large.

In binary self-synchronizing stream ciphers, low order approximations to the feedback function, as a boolean function of both the secret key and ciphertext, can also be used for the secret key reconstruction from known or chosen cipher-

text. In binary synchronous stream ciphers, every keystream bit is a boolean function of the secret key and possibly of known message key as well. Low order approximations to such boolean functions, for a suitable set of keystream bits, can be a basis for the secret key reconstruction from known keystream sequence and known or chosen message key. Another interesting objective would be to approximate every keystream bit as a low order boolean function of a number (close to the memory size of the keystream generator) of previous keystream bits, where the secret key is fixed and unknown. The resulting linear [7] (for $r = 1$) or nonlinear models of the keystream generator may then be used for the plaintext reconstruction from known ciphertext or, even, for the secret key reconstruction as well. In a structure-based approach, one may determine and use linear approximations to the next-state and output boolean functions to obtain linear sequential circuit approximations of the whole keystream generator [6], [7], which in turn can be exploited for finding linear models and for correlation attacks as well.

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. C. Ding, G. Xiao, and W. Shan, *The Stability Theory of Stream Ciphers. Lecture Notes in Computer Science*, vol. 561, Springer-Verlag, 1991.
3. G. C. Clark, Jr. and J. B. Cain, *Error-Correcting Coding for Digital Communications.* New York: Plenum Press, 1982.
4. R. Forré, "A fast correlation attack on nonlinearly feedforward filtered shift-register sequences," Advances in Cryptology – EUROCRYPT '89, *Lecture Notes in Computer Science*, vol. 434, J.-J. Quisquater and J. Vandewalle eds., Springer-Verlag, pp. 586-595, 1990.
5. R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.
6. J. Dj. Golić, "Correlation via linear sequential circuit approximation of combiners with memory," Advances in Cryptology – EUROCRYPT '92, *Lecture Notes in Computer Science*, vol. 658, R. A. Rueppel ed., Springer-Verlag, pp. 113-123, 1993.
7. J. Dj. Golić, "Linear cryptanalysis of stream ciphers," Fast Software Encryption – Leuven '94, *Lecture Notes in Computer Science*, vol. 1008, B. Preneel ed., Springer-Verlag, pp. 154-169, 1995.
8. 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*, vol. 1029, E. Dawson and J. Golić eds., Springer-Verlag, pp. 186-200, 1996.
9. T. Kasami, S. Lin, and W. W. Peterson, "New generalizations of the Reed-Muller codes, part I: primitive codes," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 189-199, Mar. 1968.
10. E. L. Key, "An analysis of the structure and complexity of nonlinear binary sequence generators," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 732-736, Nov. 1976.
11. J. L. Massey, *Threshold Decoding.* Cambridge, MA: MIT Press, 1963.

12. J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122-127, Jan. 1969.
13. M. Matsui, "Linear cryptanalysis method for DES cipher," Advances in Cryptology – EUROCRYPT '93, *Lecture Notes in Computer Science*, vol. 765, T. Helleseth ed., Springer-Verlag, pp. 386-397, 1994.
14. U. M. Maurer, "New approaches to the design of self-synchronizing stream ciphers," Advances in Cryptology – EUROCRYPT '91, *Lecture Notes in Computer Science*, vol. 547, D. W. Davies ed., Springer-Verlag, pp. 458-471, 1991.
15. W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1(3), pp. 159-176, 1989.
16. W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions," Advances in Cryptology – EUROCRYPT '89, *Lecture Notes in Computer Science*, vol. 434, J.-J. Quisquater and J. Vandewalle eds., Springer-Verlag, pp. 549-562, 1990.
17. W. Meier and O. Staffelbach, "Correlation properties of combiners with memory in stream ciphers," *Journal of Cryptology*, vol. 5(1), pp. 67-86, 1992.
18. 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.
19. 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.
20. W. Millan, "Low order approximation of cipher functions," Cryptographic Policy and Algorithms – Brisbane '95, *Lecture Notes in Computer Science*, vol. 1029, E. Dawson and J. Golić eds., Springer-Verlag, pp. 144-155, 1996.
21. R. A. Rueppel, *Analysis and Design of Stream Ciphers*. Berlin: Springer-Verlag, 1986.
22. T. Siegenthaler, "Correlation immunity of nonlinear combining functions for cryptographic applications," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 776-780, Sept. 1984.
23. T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Comput.*, vol. C-34, pp. 81-85, Jan. 1985.
24. T. Siegenthaler, "Cryptanalyst's representation of nonlinearly filtered ML-sequences," Advances in Cryptology – EUROCRYPT '85, *Lecture Notes in Computer Science*, vol. 219, F. Pichler ed., Springer-Verlag, pp. 103-110, 1986.
25. F. J. Williams and N. J. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam: North-Holland, 1988.
26. G. Z. Xiao and J. L. Massey, "A spectral characterization of correlation-immune combining functions," *IEEE Trans. Inform. Theory*, vol. IT-34, pp. 569-571, May 1988.
27. 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.
28. M. V. Živković, "On two probabilistic decoding algorithms for binary linear codes," *IEEE Trans. Inform. Theory*, vol. IT-37, pp. 1707-1716, Nov. 1991.