Efficient Approximation of Higher Order Boolean function in a Low Order Function

Mehreen Afzal[1] and Ashraf Masood[1]

College of Signals, National University of Science and Technology, Pakistan

Abstract. A few of non-linear approximation methods for Boolean functions have been developed but they are not of practical application. However, if a low order Boolean function can be found that can nearly approximate a higher order Boolean function of an encryption technique then the low order Boolean function can be used to exploit the cipher. Such a technique can become a strong cryptanalytic tool and can sneak in a cipher. In this article, an efficient method has been devised to find non-linear low degree approximation of the Boolean function. The algorithm is based on non-linear filter generator followed by solving Galois field 2 equations. To find best approximations execution time of the proposed algorithm is tremendously low as compared the brute force search. Suggested method is very efficient and of practical nature.

1 Introduction

Boolean functions have wide applications in Cryptography. A cryptographic Boolean function must be of high degree along with high non-linearity, resiliency and algebraic immunity. However, low degree approximating function of a high degree Boolean function of any cipher may threaten its security. A low order Boolean function that can well approximate a higher order Boolean function can be used to mount any cryptanalytic attack against a cipher. Linear approximations can be obtained efficiently while using algorithms based on fast Walsh transform. To find non-linear approximations in an efficient way is still an open area of research. Amongst available are these [1–3]. Most of them reveal cryptographic relevance of finding low degree approximations and offer some theoretical results without any practical application. Lower bounds of high order nonlinearities and their relation with algebraic immunity is reported in some recent researches including [4–6]; however they do not offer any method to find non-linearity profile of a Boolean function. This article is devoted to determine the same in a practical way. The proposed method is a two stage scheme, firstly, find approximations independent of correlation coefficient, secondly, getting the highest correlation coefficient amongst these approximations. The success of our algorithm in finding best approximation of the specified lower degree is demonstrated by its application on cryptographically important Boolean function of up to ten variables.

Our solution is a search method that follows the concept by Golic [1] to represent non-linear approximation as a linear recurring sequence generated by binary linear filter generators. However, instead of using the decoding procedure iteratively, we use simultaneous solution of equations. Our objective is to obtain best approximation. For Boolean functions with smaller number of variables we have verified using the brute force search that our search method can successfully find best approximation. Any method other than brute force search does not exist which can find exactly the best low order approximation. That is why we compare our method with the brute force search first to verify that our search succeeds in finding best low order approximation and another comparison is of time taken by both.

Rest of the paper is organized as follows: Section 2 is about related work, in Section 3 proposed algorithm for finding low order approximations with its performance and efficiency and some applications is discussed. The article is concluded finally in Section 4.

2 Previous Work

Maurer considered Boolean function of n variables and r degree as code words of a binary rth -order Reed Muller code (RM) with parameters $(2^n, \sum_{i=0}^{r} \binom{n}{i}, 2^{n-r})$, [2]. Thus the problem of finding best low degree approximation or higher order non-linearity of a Boolean function may be seen as decoding problem of second order Reed Muller code RM(2,m). While first order non-linearity is algorithmically related with Walsh transform, very little is known about finding higher order, even 2nd order, non-linearity. Some work can be found in [7, 8]. But these give bounds mainly for some peculiar functions in small number of variables.

Another approach for finding low order approximations given by Millan [3] is based on finding Hamming sphere sampling of order r to find an approximation of the same order. Each Hamming sphere thus finds a candidate approximation function to a given Boolean function, for each candidate the correlation coefficient is calculated and the one with maximum magnitude of the correlation coefficient is then picked as a low order approximation.

Golic [1] applied iterative error-correction algorithms to find non-linear approximation of a Boolean function, which is represented as a linear recurring sequence generated by binary linear filter generators. The procedure comprises of two steps, in first step the code words of the linear code and required parity check equations are developed. Whereas in the second step linear recurring sequence is obtained which corresponds to the approximation of the given function by using iterative probabilistic or majority-logic decoding algorithm. His solution is thus based on appropriate parity-checks that imply 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. At the end of iterations

the Hamming distance to the best low degree approximation is considerably reduced.

3 Proposed Algorithm for Low Degree Approximation

Correlation coefficient is a parameter to compare a given high degree Boolean function to its low degree candidate. The correlation coefficient between any two Boolean functions $f(x)$ and $g(x)$ of n variables is defined following [9],

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

where $X = (x_1, x_2, ..x_n)$.
Our algorithm for finding low degree approximation involves the simultaneous solution of linear equations. First step of generating equations is inspired from the method described in [1]. The low degree approximations of a Boolean function are computed irrespective of the value of correlation coefficient. Among these approximations one can find the best possible approximation based on the value of correlation coefficient.

For an n-variables Boolean function $f(X)$ of degree d, a linear feedback shift register (LFSR) of length n is considered with a primitive feedback polynomial. If $r(< d)$-degree approximations of $f(X)$ are required, then total of $K = 2^C, C = \sum_{i=0}^{r}\binom{n}{i}$, candidate Boolean functions are there. All of these K Boolean functions can be expressed as a linear combination of C Boolean monomials or product Boolean functions. Step wise execution of the proposed procedure is explained next as Algorithm 1.

Algorithm 1 Search Algorithm for non-linear approximations of Boolean function

Require: A Boolean function f of n variables and degree d, An LFSR l of length n with primitive polynomial h, A set S of C (all) product Boolean functions of degree up to r in Lex order

Ensure: A Boolean function g of degree r, $(r < d)$

1: select any initial state of LFSR l
2: Clock the LFSR C times, and obtain output bits filtered from each element of set S.
3: Generate a matrix A, $(C \times C)$. Each row of which represents the output bits generated by one element of set S
4: Generate a vector B, $(1 \times C)$, of the output bits filtered by f
5: Solve the system of equations $AX = B^T$
6: Obtain function g, using the values of X as coeff of its ANF
7: Calculate the correlation coeffiecient c between f and g

Approximating function of the required correlation coefficient can be obtained by using all possible initial states of LFSR or until no other good lower

degree approximation exists. To improve further experiment can be repeated with different feedback polynomial.

3.1 Analysis and Results

For simulation we have developed an automated procedure implemented in Maple 10 [10] to find all r-degree approximations of a given Boolean function with one primitive polynomial used as the connection polynomial and all possible initial states of the LFSR. All simulations are performed on a PC with CPU at 1.73 GHz and 1 GB RAM. To optimize the implementation a set $S' = S - \{1\}$ of product monomials is used and thus $C = \sum_{i=1}^{r} \binom{n}{i}$. Boolean functions which can be generated as a linear combination of product monomials of set S would also include all compliments of the functions obtained from the set S'.

Experiments with some small Boolean functions reveal the performance efficiency of the proposed algorithm over the brute force search. Our algorithm to find second degree approximation is executed for following (n,m,d,u)-Boolean functions (5, 1, 3, 12), (6, 1, 4, 24) and (7, 2, 4, 56). Where n is the number of variables of a Boolean function which possess resiliency of order m , algebraic degree d and non-linearity u. . Execution time of our proposed method is compared with that of brute force search for finding second degree approximation of Boolean functions of 4, 5, 6 and 7 variables, as presented in Table 1. Verifying our results for functions with larger number of variables is not possible because of the time required for brute force search, as is obvious from the table below. However, we have applied our method on Boolean functions of higher degree also which will be shown in coming section.

Table 1. A Comparison of Time to Find 2nd Degree Approximations With One Primitive Polynomial Using Proposed Algorithm and All Possible Second Degree Approximations Using Brute Borce Search of Boolean functions of 4, 5, 6 and 7 Variables

	n = 4	n = 5	n = 6	n = 7
Time to find $2^n - 1$ 2nd degree approximations with one primitive polynomial and all initial states of the corresponding LFSR using proposed heuristic	0.42 sec	1.43 sec	4.88 sec	17.97 sec
Time to find all possible 2nd degree approx using brute force search	2.09 min	7 min	1.12 hour	17,690 hours

First row of Table 1 represents the time of our proposed method when approximations are obtained with all possible initial states of an LFSR of required

length. The second row is to give the time for Brute force search. The success of our method lies in the fact that amongst $2^n - 1$ approximations obtained by one primitive polynomial, we obtain one or more approximation with highest possible correlation coefficient, as confirmed by brute force search. To make out the performance of proposed method, consider a (5, 1, 3, 12) Boolean function: $f : (x_1, x_2, x_3, x_4, x_5) \longrightarrow (x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_3 + x_2x_4 + x_3x_5 + x_1x_4 + x_2x_5 + x_1x_2x_4 + x_1x_3x_4 + x_2x_4x_5 + x_1x_3x_5 + x_2x_3x_5)$ There are a total of ($2^C, C = \sum_{i=1}^{2} \binom{5}{i}$) 32768 Boolean functions of 2nd degree which can approximate our Boolean function of degree 3. Among these, there are 26 2nd degree approximations that have 0.625 correlation coefficient with the original function obtained through brute force search. Table 2 lists all the best 2nd degree approximations of the above considered (5, 1, 3, 12)-Boolean function with 3rd column showing if it were found using proposed method or not. Following 6 primitive

Table 2. List of all 2nd degree approx of the (5, 1, 3, 12)-function with corr. coefficient 0.625 obtained by brute-force search with the detail that if our proposed method succeeds in finding it

No.	2nd degree approx.	Y/N
1	$x_3 + x_5 + x_1x_4 + x_1x_5 + x_2x_3 + x_2x_4 + x_3x_4 + x_3x_5 + x_4x_5$	Y
2	$x_3 + x_4 + x_5 + x_1x_5 + x_2x_3 + x_3x_5$	N
3	$x_2 + x_5 + x_1x_2 + x_1x_3 + x_1x_4 + x_1x_5 + x_2x_3 + x_2x_5 + x_4x_5$	Y
4	$x_2 + x_4 + x_1x_2 + x_1x_3 + x_2x_3 + x_2x_4 + x_3x_4 + x_3x_5 + x_4x_5$	N
5	$x_2 + x_3 + x_4 + x_1x_2 + x_2x_4 + x_4x_5$	N
6	$x_2 + x_3 + x_4 + x_5 + x_1x_2 + x_1x_5 + x_2x_4 + x_2x_5 + x_3x_5$	Y
7	$x_1 + x_4 + x_1x_2 + x_1x_4 + x_1x_5 + x_2x_5 + x_3x_4 + x_3x_5 + x_4x_5$	Y
8	$x_1 + x_4 + x_5 + x_1x_2 + x_1x_4 + x_3x_4$	Y
9	$x_1 + x_3 + x_1x_2 + x_1x_3 + x_1x_5 + x_2x_3 + x_2x_4 + x_2x_5 + x_3x_4$	Y
10	$x_1 + x_3 + x_4 + x_5 + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_3x_5$	N
11	$x_1 + x_2 + x_5 + x_2x_3 + x_2x_5 + x_4x_5$	Y
12	$x_1 + x_2 + x_4 + x_5 + x_1x_4 + x_2x_3 + x_2x_4 + x_2x_5 + x_3x_4$	N
13	$x_1 + x_2 + x_3 + x_1x_3 + x_1x_5 + x_3x_4$	N
14	$x_1 + x_2 + x_3 + x_5 + x_1x_3 + x_2x_5 + x_3x_4 + x_3x_5 + x_4x_5$	Y
15	$x_1 + x_2 + x_3 + x_4 + x_1x_3 + x_1x_4 + x_1x_5 + x_2x_4 + x_4x_5$	Y
16	$x_1 + x_2 + x_3 + x_4 + x_5 + x_2x_3 + x_2x_4 + x_2x_5 + x_3x_4 + x_3x_5 + x_4x_5$	Y
17	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_4 + x_2x_3 + x_2x_5 + x_3x_5$	N
18	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_3 + x_2x_4 + x_2x_5 + x_4x_5$	N
19	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_3 + x_1x_5 + x_2x_4 + x_3x_5$	Y
20	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_3 + x_1x_4 + x_2x_5 + x_3x_4$	N
21	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_3 + x_1x_4 + x_2x_4 + x_2x_5 + x_3x_5$	N
22	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_3 + x_1x_4 + x_1x_5 + x_3x_4 + x_3x_5 + x_4x_5$	Y
23	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_2 + x_1x_4 + x_2x_4 + x_3x_5$	N
24	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_2 + x_1x_4 + x_1x_5 + x_2x_4 + x_2x_5 + x_4x_5$	N
25	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_2 + x_1x_3 + x_1x_5 + x_2x_3 + x_2x_5 + x_3x_5$	Y
26	$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$	Y

polynomials are applied here: $1 + x^3 + x^5$, $1 + x^2 + x^5$, $1 + x^2 + x^3 + x^4 + x^5$, $1 + x + x^3 + x^4 + x^5$, $1 + x + x^2 + x^4 + x^5$, $1 + x + x^2 + x^3 + x^5$. Table 2 shows that in this case our heuristic succeeds in finding nearly half of the best possible 2nd degree approximations of the given Boolean function.

Table 3 presents results on some Boolean functions of 5 and 6 variables. For all Boolean functions considered here, the highest correlation coefficient is verified with brute force search. It is not possible to verify, within the feasible resources, results for Boolean functions with larger number of variables. Our experiments

Table 3. Simulation Results on Some Boolean Functions of 5 and 6 variables Using 6 Primitive Polynomials of Respective Lengths

Boolean function	Correlation coefficient of best 2nd degree approx	The number of primitive polynomials which succeed to find best approx
(5, 1, 3, 12)		
$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_3 + x_2x_4 +$ $x_3x_5 + x_1x_4 + x_2x_5 + x_1x_2x_4 +$ $x_1x_3x_4 + x_2x_4x_5 + x_1x_3x_5 + x_2x_3x_5$	0.625	6
$x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_1x_5 + x_1x_3 +$ $x_2x_4 + x_3x_5 + x_1x_4 + x_2x_5 + x_1x_2x_4 + x_1x_3x_4$ $+x_2x_4x_5 + x_1x_3x_5 + x_2x_3x_5$	0.625	6
$x_1 + x_2 + x_3 + x_4 + x_5 + x_1x_2 + x_2x_3 + x_3x_4 +$ $x_4x_5 + x_1x_5 + x_1x_2x_3 + x_2x_3x_4 +$ $x_3x_4x_5 + x_1x_4x_5 + x_1x_2x_5$	0.625	6
(6, 1, 4, 24)		
$x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_6 + x_1x_6 +$ $x_1x_4 + x_2x_5 + x_3x_6 + x_1x_2x_4 + x_2x_3x_5 +$ $x_3x_4x_6 + x_1x_4x_5 + x_1x_3x_5 + x_2x_5x_6 +$ $x_1x_3x_6 + x_1x_3x_4 + x_2x_4x_5 + x_2x_4x_6$ $+x_3x_5x_6 + x_1x_4x_6 + x_1x_2x_5 + x_2x_3x_6 +$ $x_1x_2x_3x_4 + x_2x_3x_4x_5 + x_3x_4x_5x_6 +$ $x_1x_4x_5x_6 + x_1x_2x_5x_6 + x_1x_2x_3x_6 + x_1x_2x_4x_5$ $+x_2x_3x_5x_6 + x_1x_3x_4x_6$	0.75	6
$x_1x_3 + x_2x_4 + x_3x_5 + x_4x_6 + x_1x_5 + x_2x_6 +$ $x_1x_4 + x_2x_5 + x_3x_6 + x_1x_3x_4 + x_2x_4x_5 +$ $x_3x_5x_6 + x_1x_4x_6 + x_1x_2x_5 + x_2x_3x_6 +$ $x_1x_2x_4x_5 + x_2x_3x_5x_6 + x_1x_3x_4x_6$	0.625	4
$x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_5x_6 + x_1x_6$ $+x_1x_3 + x_2x_4 + x_1x_4 + x_3x_5 + x_4x_6 + x_1x_5 + x_2x_6 +$ $x_2x_5 + x_3x_6 + x_1x_3x_4 + x_2x_4x_5 +$ $x_1x_2x_3 + x_2x_3x_4 + x_3x_4x_5 + x_4x_5x_6$ $+x_1x_5x_6 + x_1x_2x_6 + x_3x_5x_6 + x_1x_4x_6 + x_1x_2x_5 +$ $x_2x_3x_6 + x_1x_2x_3x_4 + x_2x_3x_4x_5$ $+x_3x_4x_5x_6 + x_1x_4x_5x_6x_1x_2x_5x_6 + x_1x_2x_3x_6$ $+x_1x_2x_4x_5x_2x_3x_5x_6 + x_1x_3x_4x_6$	0.5	6

include rotation symmetric Boolean functions of 5, 6 and 7 variables, which have many desirable cryptographic characteristics [11], [12]. In Table 3, results on a few functions are listed.

We have applied our algorithm to obtain 2nd and 3rd degree approximations of Boolean functions up to 15 variables. For larger $n(> 13)$, time to find approx-

imations with the proposed algorithm becomes beyond feasibility. For instance, for $n = 17$, almost 2^{13} second degree approximations can be determined with our method with one primitive polynomial in ≈ 23 hours. Although, the time required to determine $2^{17} - 1$ approximations, with one primitive polynomial, still remains very low than the brute force search which will involve computations of 2^{153} Boolean functions. Increase in the time with the increase in number of variables is presented Figure 1.

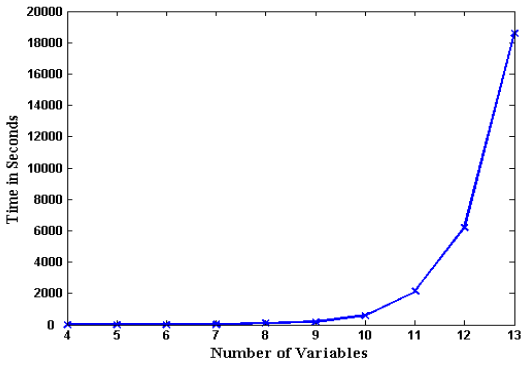

Fig. 1. Time taken by proposed algorithm to find 2nd degree approximations of Boolean functions upto 13 variables

4 Some Applications of the Proposed Method

Experiments with the proposed algorithm are performed within the computation resources mentioned in Section 3.1. With better available resources these results can be further improved. In this section some possible applications of the proposed method are discussed.

Relevance of the proposed method can be found in a number of applications. First of all it can be efficiently used for the analysis of cryptographically significant Boolean functions. For different properties of Boolean functions like non-linearity, high degree, resiliency and algebraic immunity different constructions have been proposed, mentioning a few: [13–23, 12]. We experiment with (7, 2 ,4, 56) and (8, 1, 6, 116) functions including the rotation symmetric Boolean functions (RSBF) [21, 11], and construction for maximally algebraic immune Boolean functions given in [23]. Results show that RSBFs have much higher 2nd order non-linearity than a Boolean function constructed as [23], although their other properties are comparable. Table 4 presents results of our experiments on some cryptographically significant Boolean functions.

We also find 2nd degree approximations of some higher degree Boolean functions constructed following [23] and find that even a 10 variable function of 8 degree with maximum possible algebraic immunity has 2nd degree approximation

Table 4. Application of the proposed algorithm on some cryptographically significant Boolean functions.

Boolean function	2nd Order non-linearity as obtained from the proposed method
(6,1,4,24) RSBF	8 to 16
(6,1,4,20) with max AI [23]	4
(7,2,5,56)	28 to 36
(7,2,4,40) with max AI [23]	8

with correlation coefficient approximately 0.5. Although most of these functions satisfy the bounds for algebraic immunity and higher order nonlinearities but still one is better than the other and this can be decided only if we can practically find these non-linearities or in other words lower order approximations of Boolean functions. Table 5 summarizes our results.

Table 5. Results on Boolean functions constructed following [23]

Boolean function	Correlation Coefficient of 2nd degree approximation
(6,1,4,20)	0.875
(7,2,4,40)	0.875
(8,1,5,88)	0.625
(8,3,4,80)	0.875
(9,2,5,176)	0.5
(10,1,8,372)	0.429

Use of linear approximations in cryptanalysis has been widely studied. However a few evidence of actual use of non-linear approximations in cryptanalysis is reported because of the difficulty in finding good non-linear lower degree approximations in a practical way. Following the initial linear attacks against DES [24], some improvements can be found including [25–28]. Later much flexibility in linear attack is shown by Knudsen and Robshaw [29] where they discussed the possibility of replacing linear approximations with non-linear approximations.

Algebraic attacks on stream ciphers have gained much attention in recent years soon after their emergence in [30, 31]. Though later on Algebraic attack on stream ciphers mostly uses the idea of annihilator [30–32], but earlier it also emerged from the idea of using low order non-linear approximations in a type of correlation attack [33]. So this kind of Algebraic attack with low order non-linear approximations is still possible for ciphers involving any Boolean functions if these approximations can be found efficiently and in a practical way.

5 Conclusion

A fast and practical search method based on non-linear filter generator and simultaneous solution of equations is presented to determine low degree non-linear approximations of a Boolean function. For functions with small number of variables, the efficiency of our proposed method over the brute force search

of non-linear approximations is verified experimentally. Performance of the proposed method lies in its success in finding best approximation or higher order nonlinearity of a given Boolean function. Some applications are also discussed to highlight the importance of actually finding the approximation in a practical manner.

References

1. Golic, J.D.: Fast low order approximation of cryptographic functions. In: Advances in Cryptology-Eurocrypt'96. Volume 1070., Springer-Verlag (1996) 268–282
2. Maurer, U.M.: New approaches to the design of self-synchronizing stream ciphers. In: Advances in Cryptology-Eurocrypt'91. Volume 547., Springer-Verlag (1991) 458–471
3. Millan, W.: Low order approximation of cipher functions. In: Cryptography: Policy and Algorithms. Volume 1029., Springer-Verlag (1995) 144–155
4. Carlet, C.: On the higher order nonlinearities of algebraic immune functions. In: CRYPTO '06. Volume 4117., Springer-Verlag (2006) 584–601
5. Carlet, C., Mesnager, S.: Improving the upper bounds on the covering radii of binary reed-muller codes. IEEE Transactions on Information Theory **53**(1) (2007) 162–173
6. Claude Carlet, Deepak Kumar Dalai, K.C.G., Maitra, S.: Algebraic immunity for cryptographically significant boolean functions: Analysis and construction. IEEE Transactions on Information Theory **52**(7) (2006) 3105–3121
7. Kabatiansky, G., Tavernier, C.: List decoding of second order reed-muller codes. (2005)
8. Dumer, I., Kabatiansky, G., Tavernier, C.: List decoding of second order reed-muller codes up to the johnson bound with almost linear complexity, IEEE Press (2006) pp. 138–142
9. Rueppel, R.A.: Analysis and design of stream ciphers. Springer-Verlag New York, Inc., New York, NY, USA (1986)
10. Maple: (Mathematics and engineering software)
11. Stanica, P., Maitra, S.: Rotation symmetric boolean functionscount and cryptographic properties. Discrete Applied Mathematics **ArticlesInPress** (Accepted Dec 1, 2006)
12. Maximov, A., Hell, M., Maitra, S.: (Plateaued rotation symmetric boolean functions on odd number of variables)
13. Sarkar, P., Maitra, S.: Construction of nonlinear boolean functions with important cryptographic properties. In: Advances in Cryptology-Eurocrypt'00. Volume 1807., Springer-Verlag (2000) 485–506
14. Sarkar, P., Maitra, S.: Nonlinearity bounds and constructions of resilient boolean functions. In: Advances in Cryptology-Crypto'00. Volume 1880., Springer-Verlag (2000) 515–532
15. Clark, J.A.: Almost boolean functions: The design of boolean functions by spectral inversion. Computational Intelligence **20** (August 2004) 450–462(13)
16. Meier, W., Staffelbach, O.: Nonlinearity criteria for cryptographic functions. In: EUROCRYPT'89. Volume 4234., Springer-Verlag (1989) 549–562
17. Pasalic, E., Johansson, T., Maitra, S., Sarkar, P.: New constructions of resilient and correlation immune boolean functions achieving upper bounds on nonlinearity (2001)
18. Tarannikov, Y.: On resilient boolean functions with maximal possible nonlinearity. In: Progress in Cryptology-Indocrypt'00. Volume 1977., Springer-Verlag (2000) 19–30
19. Maitra, S., Pasalic, E.: Further constructions of resilient boolean functions with very high nonlinearity. IEEE Transactions on Information Theory **48**(7) (2002) 1825–1834
20. Sarkar, P., Maitra, S.: (Highly nonlinear balanced boolean functions with important cryptographic properties)

21. Stanica, P., Maitra, S., Clark, J.A.: Results on rotation symmetric bent and correlation immune boolean functions. In: Fast Software Encryption-FSE'04. Volume 3017., Springer-Verlag (2004) 161–177
22. Dalai, D.K.: On Some Necessary Conditions of Boolean Functions to Resist Algebraic Attacks. PhD thesis, Indian Statistical Institute, Kolkata, India (2006)
23. Deepak Kumar Dalai, K.C.G., Maitra, S.: Cryptographically significant boolean functions: Construction and analysis in terms of algebraic immunity. In: Fast Software Encryption-FSE'05. Volume 3557., Springer-Verlag (2005) 98–111
24. Matsui, M.: Linear cryptanalysis method for des cipher. In: EUROCRYPT '93: Workshop on the theory and application of cryptographic techniques on Advances in cryptology. Volume 765. (1994)
25. Burton S. Kaliski, J., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations. In: CRYPTO '94: Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology. Volume 839., London, UK, Springer-Verlag (1994) 26–39
26. Jr., B.S.K., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations and feal. In: Fast Software Encryption-FSE'95. Volume 1008. (1994) 249–264
27. Tokita, T., Sorimachi, T., Matsui, M.: Linear cryptanalysis of loki and s^2des. In: Advances in Cryptology-Asiacrypt'94. Volume 917., Springer-Verlag (1994) 293–303
28. Youssef, A., Tavares, S., Mister, S., Adams, C.: the linear approximation of injective s-boxes (1995)
29. Knudsen, L.R., Robshaw, M.J.B.: Non-linear approximations in linear cryptanalysis. LNCS **1070** (1996) 224–236
30. Courtois, N., Meier, W.: Algebraic attacks on stream ciphers with linear feedback. In: Advances in Cryptology-Eurocrypt'03. Volume 2656., Springer-Verlag (2003) 345–359
31. Armknecht, F., Krause, M.: Algebraic attacks on combiners with memory. In: Advances in Cryptology-Crypto03. Volume 2729., Springer-Verlag (2003) 162–175
32. Courtois, N.: Fast algebraic attacks on stream ciphers with linear feedback. In: Advances in Cryptology-Crypto03. Volume 2729. (2003) 176–194
33. Courtois, N.: Higher order correlation attacks, xl algorithm and cryptanalysis of toyocrypt. In: ICISC. Volume 2587. (2002) 182–199