

Research on Optimization Strategy of Construction Methods of Boolean Functions

Chunlin E*, Tao Zhang*, Bin Wang*

*School of Computer Science and Technology, Fudan University, No.220, Handan Rd., Shanghai, China

09210240008@fudan.edu.cn, 09210240051@fudan.edu.cn, bwang@fudan.edu.cn

Abstract— Considering the algebraic immunity (AI) as one of the most important characteristics that evaluates the resistance against algebraic attacks, the construction methods of Boolean functions which have good performance both on algebraic and other form attacks have received a lot of attention in the cryptographic research area. In this paper, we study certain properties of Boolean functions especially their algebraic immunity. Next, we analyse some exiting construction methods and show that the most important process of the whole construction is the searching course. Further, we give some specific optimization strategies of the construction of Boolean functions and make a comparison between the performances of these methods.

Keywords— Algebraic Immunity, Boolean function, algebraic attack, optimization, genetic algorithm

I. INTRODUCTION

Boolean functions with multifarious cryptographic characters are used as the keys which are very common in both stream cipher and block cipher system to encrypt the original messages. Therefore, the design of Boolean functions would be an immediate cause of the security of the cryptographic systems. To evaluate the resistance against various attacks, people have proposed some properties of Boolean functions such as nonlinearity, resilience, correlation immunity and balance degree. Since algebraic attack has become a useful analysing method to many kinds of ciphers especially LFSR-based (Linear Feedback Shift Register-based) stream ciphers, An Braeken[1], and Claude Carlet[2], who proposed the conception of algebraic immunity (AI), offered a different angle of view to analyse the security and robustness of Boolean functions that used in cryptographic systems.

However, having a high algebraic immunity is not sufficient enough for some specific kinds of algebraic attack. A fast algebraic attack (FAA) which has been introduced by Courtois in [3] has given such an example. Consequently, a good construction method of Boolean functions should have not only the maximum algebraic immunity but also other cryptographic characters such as high nonlinearity and so forth. In this paper, we researched the algebraic immunity and other features of Boolean functions, and we assayed several construction methods. The related work can be referred to [4], [5], [6]. Meanwhile, we focused on several optimization

strategies of construction methods and made a comparison between them.

II. PROPERTIES OF BOOLEAN FUNCTIONS

A Boolean function f with n variables can be seen as a mapping from F_2^n to F_2 , the finite field with two elements. We define B_n as a set of all n -variables Boolean functions. One of the standard representations of a Boolean function $f(x_1, \dots, x_n)$ is by the output column of its truth table. We call the function is balanced when there are equal number of 0's and 1's.

Another representation that is unique is called the algebraic normal form (ANF) as a multivariate polynomial over F_2 which is defined by the equation below,

$$f(x_1, \dots, x_n) = a_0 + \sum_{1 \leq i \leq n} a_i x_i + \sum_{1 \leq i < j \leq n} a_{ij} x_i x_j + \dots + a_{12 \dots n} x_1 x_2 \dots x_n$$

where the coefficients $a_0, a_i, a_{ij}, \dots, a_{12 \dots n}$ belongs to $\{0, 1\}$.

A. Nonlinearity and Algebraic Immunity

The characteristics of Boolean functions, ones we consider here mainly, with resistance against algebraic attacks are the nonlinearity and algebraic immunity.

Before presenting the definition of nonlinearity, we give the Hamming weight as one property, which is defined as the number of 1s in a binary string S . The Hamming distance between two strings, S_1 and S_2 is denoted by $d(S_1, S_2)$, and is the Hamming weight of $S_1 \oplus S_2$. The nonlinearity of an n -variable function f , i.e.

$$nl(f) = \min_{g \in A_n} (d(f, g))$$

is the distance from the set of all n -variable affine functions [2].

Definition 1: Given $f \in B_n$, define

$$AN(f) = \{g \in B_n \mid f * g = 0\}.$$

Any function $g \in AN(f)$ is called an annihilator of function f , where f and g are multivariate polynomial over F_2 and $f * g$ is the polynomial product on the field. The sufficient and necessary condition that a Boolean function has good algebraic immunity is that f and $f+1$ do not admit nonzero annihilators of low degrees.

Definition 2: Given $f \in B_n$, the algebraic immunity is defined as the minimum degree of all nonzero annihilators of f or $f+1$, and is denoted by $AI_n(f)$, i.e.

$$AI_n(f) = \min \{ \deg g \mid 0 \neq g \in B_n, fg = 0 \text{ or } (f+1)g = 0 \}.$$

To ensure enough resistance against algebraic attacks of a Boolean function, the $AI_n(f)$ of one need to be high. Also high nonlinearity is necessary for a Boolean function to withstand linear and correlation attacks. Furthermore, it is proved in [4] that, given an arbitrary n -variable Boolean function f , it is always possible to find a Boolean function g with an algebraic degree on more than $\left\lceil \frac{n}{2} \right\rceil$, which let the

degree of f^*g is $\left\lceil \frac{n}{2} \right\rceil$ at most. As a result, since we have $f(1+f)=0$, we can get $AI_n(f) \leq \deg(f)$, and $AI_n(f) \leq \left\lceil \frac{n}{2} \right\rceil$ further.

B. Symmetric Boolean Functions with Optimal AI

Any n -variable Boolean function can be represented by a unique vector $\alpha = \{\alpha_0, \alpha_1, \dots, \alpha_{2^n-1}\}$ where $\alpha_i \in GF(2), 0 \leq i \leq 2^n-1$. Symmetric Boolean functions have a kind of concise representation which is called Simplified ANF (SANF). Defined σ_i as homogeneous symmetric functions, one of which contains all the terms with degree i , where $0 \leq i \leq n$. Therefore, SANF are polynomials whose basic elements are σ_i over field $F_2[x_0, \dots, x_{n-1}]/(x_0^2 - x_0, \dots, x_{n-1}^2 - x_{n-1})$, i.e.,

$$f(\bar{x}) = \bigoplus_{i=0}^n \lambda_f(i) \sigma_i, \lambda_f(i) = \sum_{k \preceq i} v_f(k),$$

where $0 \leq i \leq n$. The vector $\lambda_f = (\lambda_f(0), \dots, \lambda_f(n))$ is called SANF vector.

Symmetric Boolean functions are a kind of Boolean functions which are special and fairly important class. They have good cryptographic properties. And the most important character is that they can be entirely described by an $(n+1)$ -bit vector, which reduces the amount of memory required for storing the function. For this reason, they are easy to be implemented in software applications and also, feasible to be optimized with some optimization algorithms in order to obtain good calculating performance. Consequently, it is of great importance to find symmetric Boolean functions with significant cryptographic properties and of realistic significance to make comparison between these optimization methods.

III. OPTIMIZATION STRATEGIES OF CONSTRUCTION

Construction of Boolean functions with good cryptographic properties can be seen as an optimal searching process on the set of vectors which are equivalent to Boolean functions. Optimization algorithms are de facto good choices on improving searching performance. There are several algorithms are extensively used in many researches related to

find optimal selections such as Simulated Annealing (SA), Genetic Algorithm (GA), Ant Colony Optimization (ACO) and so forth. We have combined some of them to the searching process of symmetric Boolean functions with high algebraic immunity and high nonlinearity in order to get better performance in the constructions. In this paper, we presented the genetic algorithm and simulated annealing optimization strategies of construction of Boolean functions, and made comparison between them.

A. Optimization with Genetic Algorithm

Genetic algorithm is an optimization strategy used in many kinds of applications, which is a simulation of Darwinian biological evolution and genetic mechanism of natural selection process and is proposed by J. Holland in 1975. Genetic algorithm has provided a general framework to solve complex system problems because the searching and calculating would not depend on gradient information or other auxiliary knowledge but just influence the object functions and corresponding fitness functions.

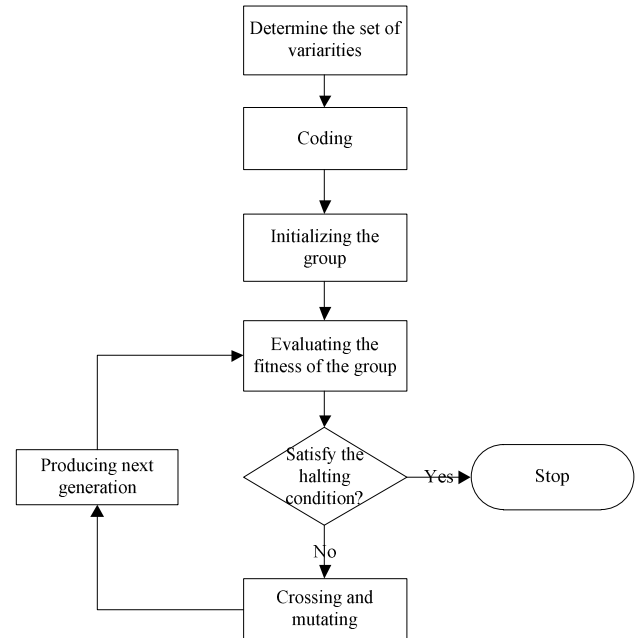


Figure 1. Process of a general Genetic Algorithm

For the purpose of obtaining a kind of Boolean functions with both high algebraic immunity and nonlinearity, we selected the searching set of symmetric Boolean functions with maximum AI. Since the symmetric Boolean functions can be represented by a vector of $n+1$ length, the searching space will be diminished fairly. The process has been divided into four steps as shown in Figure 1, namely initialization of groups, fitness calculation, crossover and mutation, and production of next generation of Boolean functions.

1) Selecting Initialization of Groups: We choose the set of vectors V using the theorem below randomly. The size of the group is k .

Theorem 1: Over the field F_2^n , the symmetric function f has maximum algebraic immunity with the value vector as,

$$V_f(i) = \begin{cases} 0 & i < \left\lceil \frac{n}{2} \right\rceil \\ 1 & \text{else} \end{cases} \quad (1).$$

Especially, when n is even, there would be a class of symmetric Boolean functions with maximum algebraic immunity which could be the initialization groups in our method.

2) Fitness Calculating: Calculation of fitness is the most important section in the iteration process, because the fitness evaluation function will influence the construction results. Considering the initial group we chose, the nonlinearity here must be the most crucial standard to evaluate the algebraic security of functions. The evaluation process can be either calculating the nonlinearity directly or computing all the annihilators of minimal degree of the polynomial f . As a result, the higher nonlinearity a function has, the more possibly it lives.

After evaluating the functions, we chose m individuals with largest nonlinearity to be the initial individuals of next generation. Our experiments showed that it had best results when m was around 5.3% of the group size k .

3) Crossover and Mutation: Crossover and mutation operations are easy here because the chromosomes are the vectors which only contain 0s and 1s. However, after every step of mutation, the method should test whether the output chromosomes are symmetric or not. If they are, the mutation results can be reserved or be discarded on the contrary. The experiment indicated that neither too high nor too low the probability will get good result after 100 generations. The nonlinearity is a relative value as we defined the best result as 1.00.

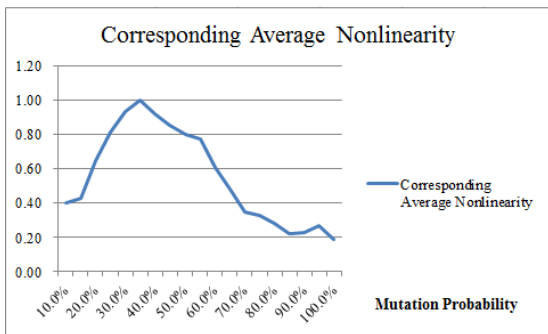


Figure 2. The average nonlinearity corresponds to mutation probability.

4) Halting Conditions: For terminating the method, two independent halting conditions are given. The first one is to prescribe a limit number of iterations. And the second one is to halt the algorithm when the average nonlinearity has reached the standard of bent functions, that is $2^{n-1} - 2^{n/2-1}$, or the average nonlinearity has not been changed for several generations.

B. Optimization with Simulated Annealing

Simulated annealing (SA) is a generic probabilistic heuristic method for the global optimization problem of a given function in a large search space. The name and inspiration of this algorithm, which was described firstly by Scott Kirkpatrick, come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects.

In order to apply the SA method to the construction of symmetric Boolean functions, the following parameters are needed: the state space, the energy function, the candidate generator procedure, the acceptance probability function, and the annealing schedule with initial temperature. These choices can have a significant impact on the method's effectiveness.

1) Candidate Generator Procedure and State Space: The method is similar to genetic algorithm so that we chose a class of symmetric Boolean functions with maximum algebraic immunity as the state space. The candidate generator procedure function can use the uniform probability distribution to choose candidate solution in the state space randomly.

2) Acceptance Probability Function: After iterations of the SA method, the current state is expected to have much lower energy than a random state. Consequently, we should skew the generator towards candidate moves where the energy of the destination states is likely to be similar to that of the current state. The main principle we use here is Metropolis–Hastings algorithm which shows that the probability p to move is defined as,

$$p = \begin{cases} 1 & \text{if } E(X_{new}) < E(X_{old}) \\ \exp\left(-\frac{E(X_{new}) - E(X_{old})}{T}\right) & \text{if } E(X_{new}) \geq E(X_{old}) \end{cases}.$$

3) Annealing Schedule: We define that $T(t)$ is the temperature at time t . In this optimization method, we used the classic annealing schedule function

$$T(t) = \frac{T_0}{\lg(1+t)},$$

where T_0 is the initial temperature which is average nonlinearity of several Boolean functions chosen randomly in the state space.

4) Terminating Principle: Different from the halting condition of genetic algorithm, there are two terminating principles which are inner loop terminating condition and outer loop terminating condition respectively. The inner condition is usually based on whether the average nonlinearity has been changed for several iterations or not. However the outer one can either use the limit number of iterations or the threshold temperature.

In fact, how to choose the parameters of simulated annealing algorithm is a very complicated problem, because there is no standard. The parameters can be amended after more experiments. Also, there are some other conditions and

circumstances should be considered carefully such as the restarting problems and the choosing of initial temperature and terminating principles, which are worth discussing more specifically in further work.

IV. COMPARISON BETWEEN TWO STRATEGIES

We have done some experiments on the two optimization strategies of construction of Boolean functions and made a simple comparison between these two methods.

The strategy based on genetic algorithm has an easier implementation and fewer restrictions than the one based on simulated annealing algorithm. However, GA based method seems to be more sensitive to the initial parameters than the other one, which can always get global optimal solution theoretically. Additionally, as we mentioned, there are several problems which are worth to pay more attention in SA based strategy such as the restarting and so forth.

V. CONCLUSIONS

Cryptography is the most important branch and a tool of information security. The study of algebraic attacks shows a new direction of the development of cryptography. In this paper, we introduced the algebraic immunity of Boolean functions and other characteristics withstanding the algebraic attacks. Meanwhile, we analysed the methods of constructing Boolean functions with optimal algebraic immunity and other cryptographic properties. At last, the processes of two optimization strategies of Boolean functions' construction are proposed and a simple comparison between them has been made.

REFERENCES

- [1] A. Braeken and B. Preneel. On the Algebraic Immunity of Symmetric Boolean Functions, *Progress in Cryptology - Indocrypt 2004*, volume 3797 of Lecture Notes in Computer Science, pp .35-48.
- [2] C. Carlet, D. K. Dalai, K. C. Gupta and S. Maitra. Algebraic Immunity for Cryptographically Significant Boolean Functions: Analysis and Construction, *IEEE Transactions on Information Theory*, 2006, pp. 3105-3120.
- [3] N. Courtois. Fast algebraic attacks on stream ciphers with linear feedback, *Advances in Cryptology-CRYPTO 2003*, Lecture Notes in Computer Science, vol. 2729 pp. 176-194.
- [4] J. Peng and Haibin Kan, Annihilators and algebraic immunity of symmetric Boolean functions, *IEICE Transactions on Fundamentals*, Vol.E94-A, No.6, Jun. 2011.
- [5] J. Peng, Quanshui Wu, and Haibin Kan. On Symmetric Boolean Functions With High Algebraic Immunity on Even Number of Variables, *IEEE Transactions on Information Theory*, vol. 57, No. 10, Oct, 2011.
- [6] Yuan Li, Min Yang, Haibin Kan, Constructing and Counting Boolean Functions on Even Variables with Maximum Algebraic Immunity. *IEICE Transactions on Fundamentals*, Vol.E93-A, No.3, Mar. 2010.
- [7] F. Armknecht, C. Carlet, P. Gaborit. Efficient Computation of Algebraic Immunity for Algebraic and Fast Algebraic Attacks. *Advances in Cryptology*, 2006
- [8] Yuan Li, Haibin KAN, and Kokichia, Futatsugi, Note on "On the Construction of Boolean Functions with Optimal Algebraic Immunity", *IEICE Transactions on Fundamentals*, Vol.E94-A, No.9, Sep. 2011.
- [9] J. Peng and Haibin Kan, Constructing correlation immune symmetric Boolean functions, *IEICE Transactions on Fundamentals*, Vol.E94-A, No.7, Jul. 2011.
- [10] H. Wang, J. Peng, Y. Li, and Haibin Kan, On 2k-Variable Symmetric Boolean Functions with Maximum Algebraic Immunity k, submitted to *IEEE Transactions on Information Theory*, under the second review.
- [11] Q. Wang, J. Peng, Haibin Kan and X. Xue, Constructions of cryptographically significant Boolean functions using primitive polynomials, *IEEE Transactions on Information Theory*, Vol. 56, No.6, June 2010.
- [12] X. Zhang, Q. Wang, B. Wang, and Haibin Kan, A Constructive Method of Algebraic Attack with Less Keystream Bits Date of Evaluation, *IEICE Transactions on Fundamentals*, Vol.E94-A, No.10, Oct. 2011.
- [13] Q.C. Wang, X.Y. Xue and Haibin Kan, New Balanced Boolean Functions with Good Cryptographic Properties, *IEICE Transactions on Fundamentals*, Vol.E92-A, No.10, Oct. 2009.