Heuristic Search by Particle Swarm Optimization of Boolean Functions for Cryptographic Applications

Luca Mariot Alberto Leporati

Dipartimento di Informatica, Sistemistica e Comunicazione – Università degli Studi di Milano Bicocca, Viale Sarca 336/14, 20126 Milano, Italy
{luca.mariot, leporati}@disco.unimib.it

Abstract

We present a Particle Swarm Optimizer for generating boolean functions with good cryptographic properties. The proposed algorithm updates the particles positions while preserving their Hamming weights, to ensure that the generated functions are balanced, and it adopts Hill Climbing to further improve their nonlinearity and correlation immunity. The results of the optimization experiments for $n = 7$ to $n = 12$ variables show that this new PSO algorithm finds boolean functions with good trade-offs of nonlinearity, resiliency and Strict Avalanche Criterion.

Keywords– Particle Swarm Optimization, Boolean Functions, Cryptography, Hill Climbing

1 Introduction

It is known that, in order to withstand specific cryptanalytic attacks, the boolean functions employed in block and stream ciphers must satisfy several *cryptographic properties*.

The goal of this paper is to investigate the use of Particle Swarm Optimization (PSO) to design boolean functions featuring good combinations of cryptographic properties. The main idea is to apply Kennedy and Eberhart's discrete PSO [3] to explore the space of *balanced* boolean functions. This is achieved through a new update procedure, inspired by permutation PSO [2], which preserves the Hamming weight (i.e., the number of nonzero coordinates) of the particle positions. Further, the proposed PSO algorithm is coupled with an Hill Climbing technique [4] to locally improve the nonlinearity and deviation from correlation immunity of the particles.

2 Boolean Functions

We briefly recall the cryptographic properties of boolean functions considered in our paper. For further details, see [1].

A *boolean function* of n variables is a mapping of the form $f : \mathbb{F}_2^n \to \mathbb{F}_2$, represented either as a 2^n-bit vector encoding its *truth table* or as a multivariate polynomial called the *Algebraic Normal Form* (ANF) of f. Function f is *balanced* if its truth table is composed by an equal number of 0s and 1s, while the *algebraic degree* $deg(f)$ of f is the degree of its ANF. The *nonlinearity* of f, denoted by $Nl(f)$, is the minimum Hamming distance of f from the set of *affine functions* (i.e., functions of degree 1). Further, f is *k-th order correlation immune* ($CI(k)$) if, by fixing i input variables for $1 \leq i \leq k$, the truth tables of the resulting restrictions of f all have the same Hamming weight. A function which is both balanced and $CI(k)$ is *k-resilient*. Finally, function f satisfies the *Strict Avalanche Criterion* (SAC) if by complementing a single input bit the value of f changes with probability 1/2, and the *absolute indicator* AC_{max} of f is the maximum absolute value of its *autocorrelation function*.

Boolean functions used in symmetric ciphers should have high algebraic degree and nonlinearity, satisfy resiliency of high order and the SAC, and have low absolute indicator. There are however some trade-offs among these criteria: for example, given $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *Siegenthaler's bound* states that $deg(f) \leq n-k-1$, while for *Tarannikov's bound* it results that $Nl(f) \leq 2^{n-1} - 2^{k+1}$, where k is the resiliency order.

3 PSO Algorithm

The baseline algorithm adopted in our experiments is Kennedy and Eberhart's discrete PSO [3], where the positions of the particles are 2^n-bit vectors representing the truth tables of boolean functions of n variables. In the basic version of discrete PSO, the velocity vector of a particle specifies for each coordinate of its position vector the *probability* that the corresponding bit flips its value. Since each bit is updated independently from the others, this procedure does not guarantee that the generated boolean function is balanced.

To solve this drawback, we designed a new *swap-based* operator inspired from Hu, Eberhart and Shi's permutation PSO [2]. Given the balanced binary vector $x \in \mathbb{F}_2^{2^n}$ and the corresponding probability vector $p \in [0,1]^{2^n}$, for each coordinate $i \in \{1, \cdots, 2^n\}$ a swap is performed with probability p_i as follows. First, the value of x_i is compared with that of the *global best solution* g found by the whole swarm at the same index, g_i. If they are equal, then no action is taken. Otherwise x_i is swapped with x_k, where $k \neq i$ is such that $x_k \neq g_k$ and $x_k \neq x_i$. In this way, the Hamming distance of vector x from the global best g is decreased by 2, while its balancedness is preserved. The whole update process is then repeated using the *local best solution* of the particle. The velocity vector of a particle is in turn updated using the classical PSO velocity equation, which is normalised through the *logistic function* to get meaningful probability values.

Table 1: CGA-Evolved PSO Parameters

fit_j	w	φ	ψ	v_{max}
fit_1	0.5067	2.8751	1.3587	3.5008
fit_2	0.7614	2.0073	2.0273	2.7183
fit_3	0.2828	2.1824	0.8951	4.2639

We tested our Particle Swarm Optimizer with three fitness functions, all of which are maximised. Given $f : \mathbb{F}_2^n \to \mathbb{F}_2$, the fitness functions are defined as follows:

$$fit_1(f) = Nl(f) - [CID_1(f)/4] - [SACD(f)/8]$$
$$fit_2(f) = Nl(f) - CID_2(f)$$
$$fit_3(f) = Nl(f) - AC_{max}(f)$$

where $CID_1(f)$, $CID_2(f)$ and $SACD(f)$ respectively represent the *deviations* of f from $CI(1)$, $CI(2)$ and the SAC as defined in [4]. A deviation value of 0 means that the function satisfies the corresponding cryptographic property.

To further improve the performance of our PSO algorithm, after the application of the position update operator each particle is optimized with the Hill Climbing (HC) algorithm designed by Millan, Clark and Dawson [4], which swaps a pair of bits in the truth table of a boolean function to increase its nonlinearity and decrease its deviation from $CI(k)$.

4 Experimentation

Recall that the PSO velocity equation depends on four parameters: the *inertia* parameter w, the *social* and *cognitive constants* φ and ψ, which respectively determine the influence of the global best and the local best solution, and the *maximum velocity value* v_{max}. For each fitness function we tuned these parameters using a *Continuous Genetic Algorithm* (CGA). Table 1 reports the values of the best CGA-evolved parameters adopted for our experiments.

We tested our PSO algorithm on the spaces of balanced boolean functions from $n = 7$ to $n = 12$ variables. The number of particles and iterations were set to $P = 200$ and $I = 400$ respectively, and for each value of n and fitness function we carried out $R = 100$ PSO runs.

Table 2 shows for each fitness function the cryptographic properties of the global best solution g which scored the highest fitness value among all the $R = 100$ PSO optimization runs. One can notice that in the case of both fit_1 and fit_2 the $CI(k)$ boolean functions discovered by PSO always reach Siegenthaler's bound, even if the algebraic degree was not evaluated by any of the three fitness functions.

Using fitness function fit_1, the best boolean functions found by our PSO algorithm are all $CI(1)$ (and thus 1-resilient, since they are also balanced), and for $n = 7$ and $n = 8$ they also satisfy the SAC.

Table 2: Best Boolean Functions Found

fit_j	Property	7	8	9	10	11	12
fit_1	Nl	56	112	236	480	972	1972
	deg	5	6	7	8	9	10
	CID_1	0	0	0	0	0	0
	$SACD$	0	0	8	8	8	8
fit_2	Nl	56	112	232	476	972	1972
	deg	4	6	7	8	9	10
	CID_1	0	8	8	8	8	16
	CID_2	0	8	8	8	8	16
fit_3	Nl	56	116	236	480	976	1972
	deg	5	6	7	9	10	11
	AC_{max}	16	32	48	80	128	208

On the other hand, our Particle Swarm Optimizer does not perform well with respect to fit_2, since for $n > 7$ the deviation from $CI(2)$ is always at least 8. However, it is worth noting that for $n = 7$ the best solution is 2-resilient and reaches both Siegenthaler's and Tarannikov's bounds.

Finally, another different behaviour of the PSO algorithm can be observed using fitness function fit_3. Indeed, one can see that as the number of variables grows the absolute indicator of the best solution gets worse. Nonetheless, for $n = 8$ and $n = 11$ the nonlinearity values achieved with fit_3 are greater than those obtained using fit_1, while they are equal in all other cases.

5 Conclusions

In this work, we applied a discrete PSO algorithm to optimize the cryptographic properties of boolean functions from $n = 7$ to $n = 12$ variables. The results of the experiments show that our PSO discovers boolean functions achieving good combinations of nonlinearity, first order correlation immunity and Strict Avalanche Criterion, while it does not perform well when it minimizes deviation from $CI(2)$ or the absolute indicator.

A possible future development on the subject is to modify the position update operator so that only the swaps which increase nonlinearity or $CI(k)$ are performed, for example by integrating the Hill Climbing algorithm directly inside the update procedure.

References

[1] C. Carlet. Boolean functions for cryptography and error-correcting codes. In Y. Crama and P. L. Hammer, editors, *Boolean Models and Methods in*

Mathematics, Computer Science, and Engineering, pages 257–397. Cambridge University Press, New York, May 2011.

[2] X. Hu, R. C. Eberhart, and Y. Shi. Swarm intelligence for permutation optimization: Case study of n-queens problem. In *Proceedings of the IEEE Swarm Intelligence Symposium, (Indianapolis, IN, April 24-26, 2003*, pages 243–246, 2003.

[3] J. Kennedy and R. C. Eberhart. A discrete binary version of the particle swarm algorithm. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, (Orlando, FL, October 12-15, 1997)*, pages 4104–4108, 1997.

[4] W. Millan and A. J. Clark. Heuristic design of cryptographically strong balanced boolean functions. In *Proceedings of EUROCRYPT'98 (Espoo, Finland, May 31–June 4, 1998), Lecture Notes in Computer Science vol. 1403*, pages 489–499, 1998.