Journal of Engineering and Applied Sciences 13 (20): 8448-8451, 2018
ISSN: 1816-949X

Design and Construction of Nonlinear Boolean Function in Stream Cipher System

Zaki S. Tywofik, Salim M. Zaki and Ali Taha Yaseen
Department of Computer Science, Dijlah University College, Baghdad, Iraq

Abstract: Nonlinear stream cipher is a fundamental system in protecting data security in computer networks due to its high level of protection. This type of cipher needs moderate level of calulcation to be attacked it by attackers. This research presents an analysis to nonlinear Boolean function and its role in nonlinear generator of sequnce keys by using generator algorithm by proposing an algorithm to generate those functions. This algorithm will be simulated to help deigners choosing the suitable permutation fucntion.

Key words: Permutation function, generator function, nonlinear function, Boolean function, stream cipher, computer networks

INTRODUCTION

Stream cipher system is vital in protecting data and ensuring security. In stream cipherpesudo-random XORed with plain-text to generate cipher text (Menezes *et al*., 2005). Stream cipher encrypts bit by bit with the use of encyption function changes over time as shown in Fig. 1.

The stream ciphers are commonly used in protecting digital communications due to its great throughput and require less complexty of hardware circuit design in addition to slight error propagation which makes it attracted much consideration (Kim *et al*., 2009; Huang *et al*., 2013). Stream ciphers are underattack using mathimatical methods which make it voulnrable and need to be secured (Liu and Lin, 2017).

Boolean functions have received attention due to its simplicity and effective computation (Kim *et al*., 2009) and play important role in designing stream cipher schemes and strengthen the security of ciphers against attacks (Zhang *et al*., 2012; Li and Zhang, 2015).

Stream cipher (use one key) similar to one-time cipher. Stream cipher uses finite sequence of generated keys saparated from each other (Zhang *et al*., 2017).

Stream cipher systems classified into two main parts: linear systems that depends on linear secret keys and non-linear that depends on nonlinear secret keys. Nonlinear stream cipher for generating secret encryption keys, depends on building nonlinear generators using two techniques, unifying and filtering. Permutation functions considered as excutable state for filtering technique. Using and generating nonlinear Boolean functions is important in desiging sequential nonlinear random generators with high complexity.

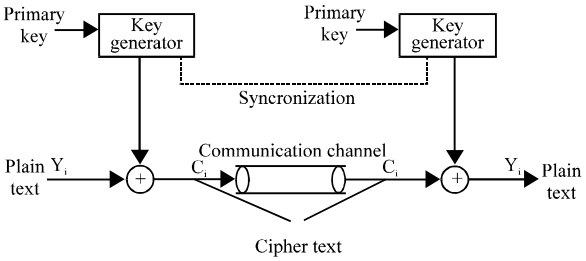

Fig. 1: Stream cipher diagram

The organization of this study is as follows. At the beginning, it presents a mathematical analysis for designing nonlinear function and premutation function by considering Boolean function used in designing stream cipher systems. Two well-known algorithms of (Piperrzik and Finkelstein, 1988) in used in the mathematical analyses proposed in this study.

MATERIALS AND METHODS

Definitions

Boolean functions: Boolean function f: $x^n \rightarrow x$ is alinear function for (n) of Boolean variables if represented as below (Menezes, 2005):

$$f(x) = f(x_1,...,x_n) = a_0 + a_1x_1 +, ..., + a_nx_n$$

where, a_0, a_i, $x_i \in$ xfor the variables i = 1, ..., n the definition of Boolean function could be generalized. The Boolean function $^nf:x^{n-}$ "xconsidered a linear function for (n) of variables if represented as:

$$f_1(x_1,...,x_n)f(x) = f(x_1,...,x_n) = f_n(x_1,...,x_n)$$

Corresponding Author: Zaki S. Tywofik, Department of Computer Science, Dijlah University College, Baghdad, Iraq

where all elements of the function $f_i(x)$ for i = 1, ..., n is a linear function links (n) of bits to one bit.

Space: Is a group of Boolean functions for (n) of variables as below:

$$F_n = (f: f:x^{n\rightarrow}xn)$$

$$L_n = (\alpha : \alpha \text{ is linear and} \alpha \in F_n)$$

The number of Linear Boolean functions within the space F_n is:

$$L_n = \left|2^{n+1}\right|$$

where, (n) is the degree of space. The following relation can determine the number of linear and nonlinear Boolean functions in space F_n:

$$F_n = 2^n$$

While the number of nonlinear Boolean function is (F_n-L_n).

Hamming weight: The weight of Boolean function $f \in F_n$ is the hamming weight of the function image f which is equal to the number of units in the truth table of function f.

Distance: Distance between Boolean functions f and g which fall in the field F_n, $[(f,g) \in F_n]$ denoted with the relation:

$$d(f,g) = wt(fi \oplus gi)$$

Where:

f (x) = $[f(x_1, ..., x_n)]$

g (x) = $[g(x_1, ..., x_n)$

Degree of nonlinear Boolean function (f): Is the shortest distance between function f and all linear functions in the same space which could be denoted in the following relation:

$$^{n-3}: Nf = \sum_{i=1/2(n-2)} 2i \; n = 2,4,6,...,$$

$$^{n-3}: 2 \sum_{i=1/2(n-3)} 2^i \; n = 3,5,7,...,$$

Permutating functions within the space F_n^n is a collection of (n) of Boolean functions in the space F_n and these functions share $[f_1, ..., f_n]$ the same weight with value wt $(f_i) = 2^{n-1}$

Example: Calculate the weight of the two linear functions $f_2 = x_1 \oplus x_2$ and $f_1 = x_1$within the space F_n. Is it possible to generate permutation for these two functions within the space F_2?

Table 1: Truth table of the Boolean functions

f_2	f_1	X_2	X_1
0	1	0	0
1	0	0	1
1	1	1	0
0	0	1	1

Solution: Table 1 represents truth table of the two functions which helps in solving the example. Noticed in Table 1 within first and second order columns which helps in producing permutation of these two functions.

Generating Boolean functions and nonlinear permutation functions: Nonlinear Boolean permutation function is build in a space based on a nonlinear Boolean function from a smallest space which is the second space F_2. With the condition that weight of this functions should be 1. Then, it expands to other spaces as in Piperrzik algorithm in 1988.

First algorithm of Piperrzik: This algorithm presents a suitable way for generating nonlinear Boolean function within a specified space based on a nonlinear functions within space F_2. This function is known as it has a high nonlinearity. Below steps describe the algorithm:

Algorithm 1; Piperrzik algorithm:

1: Randomly choosing a nonlinear function in the space F_2 and let f_i with weight equals 1.

2: Expanding the function $f^a F_i$ if (i) is an even number, the following relation is used:

$$f_{i+1} = [f_i, f_i]_{i+1} \; f_{i+1} = f_i \in F_i \quad (1)$$

3: Calculating the new function $[f_{i+1}+x_{i+1} \in F_i]$

4: Creating function f_{i+2} in space F_{i+2} considering the following relation:

$$f_{i+2} = [f_{i+1}, f_{i+1}+x_{i+1}]_{i+2} = [f_{i+1}+x_{i+2}]F_{i+2} \quad (2)$$

5: Expanding the function if (i) was an odd number, the function f_{i+1} created by the following relation:

$$f_{i+1} = [f_i, f_{i+1}+x_1]_{i+1} F_{i+1} = f_i, x_{i+1}+F_{i+1} \quad (3)$$

The algorithm continues until required space is obtained and nonlinear function want to be designed.

Design of nonlinear permutation boolean function: Not all the nonlinear Boolean functions generated by Piperrzik algorithm are nonlinear Boolean functions in the space F_n with weight of 2^{n-1}, therefore, this algorithm not suitable for generating nonlinear permutation functions. Where the weight of elements of functions $wt(f_i) = 2^{n-1}$.

Piperrzik proposed second algorithm based on designing nonlinear functions generating nonlinear permutation functions with weight of 2^{n-1} in the space F_n^n.

Second algorithm of Piperrzik: This algorithm aims to generate nonlinear permutation Boolean with weight 2^{n-1} based on designing nonlinear functions. Following steps describe the algorithm.

Algorithm 2; Piperrzik algorithm:

1: Choosing a nonlinear functions in the space F_2 and let it be f_i with weight equals to 1.

2: If (i) is an even number, the function f_{i+1} is created in the space F_i with the relation:

$$f_{i+1} = [f_i, f_i]_{i+1} = f_i + x_{i+1} \quad (4)$$

And the Boolean functions created by:

$$P_1(f_{i+1}) = f_{i+1}\,P_2(f_{i+1}) = f_{i+1} + x_1 P_3(f_{i+1}) = f_{i+1} + x_2 \quad (5)$$

3: If (i) is an odd number, the Boolean function f_{i+1} expands in the space F_{i+1} with the relation:

$$f_{i+1} = [f_i, f_i]_{i+1} = f_i \quad F_{i+1} \quad (6)$$

And the nonlinear permutation Boolean function created by:

$$P_1(f_{i+1}) = f_{i+1}\,P_2(f_{i+1}) = f_{i+1} + x_1 P_3(f_{i+1}) = f_{i+1} + x_2\,P_4(f_{i+1}) = f_{i+1} + x_3\,P_i(f_{i+1}) = f_{i+1} + x_i \quad (7)$$

The new nonlinear permutation fucntion defined as:

$$f_{i+1} = [f_i, f_i] = f_i + x_{i+1} \quad F_{i+1} \quad (8)$$

The algorithm continues until required space is obtained and nonlinear permutation function want to be designed.

Basic functions and permutation fucntions in space: Number of linear functions can be determined in the space F_n with the relation:

$$L_n = |2^{n+1}|$$

And determining the number of linear and nonlinear functions with the relation:

$$F_n = 2^{2n}$$

The number of nonlinear Boolean functions is $(F_n\text{-}L_n)$, therefore, the number of linear and nonlinear Boolean functions in the space is as follows.

RESULTS AND DISCUSSION

Second space F_2: Number of linear and nonlinear Boolean functions in this space is function as shown in Table 2 there are 8 linear Boolean functions and 8 nonlinear Boolean functions.

After executing all the linear functions in space F_2 on a computer we get (10) permutation cases out of (28) possible case as shown in Table 3.

The generated permutation cases do not have the same security level when use in generating sequence of pseudorandom key generation. Where the number of permutations can be calculated as:

$$\text{Number of permutations} = 2 = 8!/2!(8\text{-}2)! = 28$$

Table 2: Number of Boolean functions

Nonlinear functions	Linear functions
x_1x_2	0
x_1x_2	1
x_1x_2	x_1
\	
x_1x_2	-
x_1	
x_1+x_2	x_2
$\bar{x}_1+x_2$	$\bar{x}_2$
$\bar{x}_1+x_2$	$x_1 x_2$
$\bar{x}_1+\bar{x}_2$	$x_1 x_2$

Table 3: Permutation cases

f_1	f_2	Order after permutation
x_1	x_2	(1, 0, 3, 2)
x_1	x_1+x_2	(0, 1, 3, 2)
x_1	x_1+x_2	(1, 3, 2, 0)
x_2	x_1+x_2	(3, 0, 2, 1)

Table 4: Number of linear and nonlinear functions in third space

Nonlinear functions	Linear functions
$x_1\,x_2\,x_3$, $x_1\,x_2\,x_3$	0,1, x_1, x_2, x_3
$x_1\,x_2\,x_3$, $x_1\,x_2\,x_3$	x_1, x_2, x_3
$x_1\,x_2$, $x_2\,x_3$, $x_1\,x_3$	x_1+x_2, x_1+x_3
$x_1+x_2+x_3$	x_2+x_3
$x_1+x_2+x_3$	x_1+x_2, x_1+x_3
$x_1+x_2+x_3$	x_2+x_3
.	$x_1+x_2+x_3$
.	$x_1+x_2+x_3$

Table 5: Generating permutations from linear fucntions

f_1	f_2	f_3	Order after permutation
x_1	x_3	x_1+x_2	(0, 2, 4, 6, 5, 7, 1, 3)
x_1+x_2	x_3	x_1	(5, 7, 1, 3, 0, 2, 4, 6)
x_2	x_1	x_3	(5, 4, 1, 0, 7, 6, 3, 2)
x_3	x_2	x_1	(0, 4, 2, 6, 1, 5, 3, 7)

Third space F_3: The number of linea and nonlinear in third space F_3 is 256 function. The number of linear functions is 16 while nonlinear functions is 240 as in Table 4.

The number of permutation cases is 560 case. All these cases have been selected which generated 82 linear function that can generates permutation case as shown in Table 5. These permutations don have the same security level and so other spaces.

CONCLUSION

This research covered generating linear functions and nonlinear permutation functions based on Piperrzik algorithms. An analytical study also done in this research to determine the basics in the space F_2 that is used in generating permutation functions in any other space. Main points that summarize this research.

Executing Piperrzik algorithms generates functions with high nonlinearity degree using nonlinear Boolean functions within the space F_2 of (n) of variable.

It is possible to use the well-known equation to calculate the degree of nonlinearity for the Boolean function within the spaces F_2-F_4. However, when the

n = 5 finding nonlinear functions with high nonlinearity becomes inefficient because the number of functions will be (2^{32}-2^6) which considered big number and difficult to deal with it due to limitations in memory capacity at university computers. In addition to time required to run those nonlinear Boolean functions.

It is possible to generate secure permutation using linear functions in any space specially functions with XOR operation. On the other hand, nonlinear functions don have permutation unless the two algorithms of Piperrzik applied due to its support for nonlinear permutation functions with high nonlinearity.

It is recommended to choose a function with weight equal to one from the space F_2 (because nonlinearity of the function becomes at its highest level) instead of choosing a function with weight of three from the same space because its linearity degree equals zero.

REFERENCES

Huang, X., C. Wang, W. Huang and J. Li, 2013. The nonlinear filter Boolean function of LILI-128 stream cipher generator is successfully broken based on the complexity of nonlinear 01 symbol sequence. Circuits Syst., 4: 165-168.

Kim, H., S.M. Park and S.G. Hahn, 2009. On the weight and nonlinearity of homogeneous rotation symmetric Boolean functions of degree 2. Discrete Appl. Math., 157: 428-432.

Li, L.Y. and W.G. Zhang, 2015. Construction of resilient Boolean functions with high nonlinearity and good algebraic degree. Secur. Commun. Netw., 8: 2909-2916.

Liu, M. and D. Lin, 2017. Results on highly nonlinear Boolean functions with provably good immunity to fast algebraic attacks. Inf. Sci., 421: 181-203.

Menezes, A., P.V. Oorschot and S. Vanstone, 2005. Handbook of Applied Cryptography. CRC Press, Boca Raton, Florida, USA.,.

Pieprzyk, J. and G. Finkelstein, 1988. Towards effective nonlinear cryptosystem design. IEE Proc. E (Comput. Digital Tech.), 135: 325-335.

Zhang, B., X. Gong and W. Meier, 2017. Fast correlation attacks on grain-like small state stream ciphers. IACR. Trans. Symmetric Cryptology, 2017: 58-81.

Zhang, P., D. Dong, S. Fu and C. Li, 2012. New constructions of even-variable rotation symmetric Boolean functions with maximum algebraic immunity. Math. Comput. Modell., 55: 828-836.