Volume XXIII 2020
ISSUE no.1
MBNA Publishing House Constanta 2020

Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Applied study on cryptographic functions for algorithms used in communications security protocols

To cite this article: Florin Medeleanu, Narcis-Florentin Antonie,Ciprian Răcuciu, Dan Laurentiu Grecu and Florin Antohi, Scientific Bulletin of Naval Academy, Vol. XXIII 2020, pg.243-247.

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-20-I1-033

Applied study on cryptographic functions for algorithms used in communications security protocols

Florin Medeleanu[1], Narcis-Florentin Antonie[1,3], Ciprian Răcuciu[2], Dan Laurentiu Grecu[2] and Florin Antohi[1]

[1]Ministry of Defense, Strada Izvor 3-5, Sector 5, Bucharest, Romania
[2]Department of Computer Science, Titu Maiorescu University, 22 Strada Dâmbovnicului Tineretului, București 040441, Bucharest, Romania
[3]Corresponding author: narcis.antonie@yahoo.com

Abstract. Communications security is one of the most important fields to take into account when designing a system that manages information, especially when implementing such a system for the military, no matter which branch, Navy, Air Force or Army. One important field when talking about information security in general is cryptology and within cryptology linear and nonlinear Boolean functions and maps are essential, important building blocks. They are used in the design of several block and stream ciphers. The study of cryptographic properties of these functions does not only help cryptanalysis but also plays an important role in the design of cryptographic algorithms that resist well against various cryptographic attacks. Linear and differential cryptanalysis of block ciphers is mainly based on determining and exploiting linear combinations of their components. The most useful mathematical tool for studying linearity of Boolean functions is the Walsh (or Hadamard) transform. This can be regarded as a size-2 discrete Fourier transform. Another method for determining linear combinations of cipher components is that of finding and solving linear systems of equations. This article reflects the authors' effort to shed some light on this field.

1. Introduction

Cryptographic algorithms are at the core of information and data security. They are used in all sorts of applications, both civilian and military. In the military field they are extremely important due to the importance of keeping military information protected from the enemy. Cryptographic algorithms are used in all branches of the military army, air force and navy. In the navy they can be used for data transmission between ships, between ships and shore or between ships and aircrafts. There are multiple cryptographic algorithms circulating around the world right now. Many of them are used also in the military and that is why it is very important to analyse them in order to determine their characteristics. One such algorithm is the AES – Advanced Encryption Standard, which is the focus of this article.

The Advanced Encryption Standard (AES) is a cryptographic algorithm that was developed by Vincent Rijmen and Joan Daemen. This algorithm was adopted and implemented as a standard by National Institute of Standards and Technology (NIST) in 2002 [2]. A significant number of papers have demonstrated the safety of this algorithm by provable security methods or by evaluation of maximum expected differential/linear probability, but few have dealt with the practical details of linear and differential attacks on the AES [1].

In order to carry out a linear attack on this algorithm, one should determine the linear combinations of the linear part of the cipher together with the linear approximation of the non-linear part. Despite the best effort of the authors, no paper dealing with a detailed method to determine these linear combinations could be identified.

Linear cryptanalysis was first presented at the Eurocrypt conference in 1993 by M. Matsui. At the beginning, this method was described as a theoretical attack on the Data Encryption Standard cryptographic algorithm (DES) [3] and was then used successfully for the practical cryptanalysis of DES [4]. Linear cryptanalysis is a known plaintext attack which uses plaintext-ciphertext pairs in order to determine the value of the key bits [5]. The attack works on the principle of identifying "high probability occurrences of linear expressions involving plaintext bits, ciphertext bits (actually we shall use bits from the 2nd last round output), and subkey bits" [5].

2. Study of the non-linear part of an algorithm

The AES algorithm is fully described in FIPS-PUB-197 standard. In short, this algorithm has a linear and a non-linear part. The linear part is composed of *ShiftRows* and *MixColumns* transformations and the non-linear part consists of *SubBytes* transformation.

For the *S-box*, the non-linear part of the AES algorithm, it is known that the highest value of linear probability is exactly $(8/64)^2 = 2^{-6}$. This happens for five highly probable linear expression values with a bias of $\varepsilon = 2^{-4}$. In order to practically determine the highest probable values of the linear expressions for AES non-linear transformation (*S-box*), the authors used the Walsh transform. The result of the Walsh transform, Walsh spectrum, was organized as a table with rows – input linear expressions – and columns – output linear expressions. The content of the table was comprised of associated biases for each combination of input / output linear expressions.

3. Study of the linear part of an algorithm

In order to determine input / output linear expressions for the linear attack, some elements of the linear parts of the AES algorithm (e.g. *ShiftRows* transformation) don't require any specific calculation, because it is very intuitive. It is obvious that the output linear expression for *ShiftRows* transformation can be determined by simply rotating the input linear expression with 1, 2 or 3 positions (considered as Bytes). But other elements (e.g. *MixColumns* transformation) need more elaborate calculation.

The Walsh spectrum method used in Section 2 can also be used, and is most widely recommended. But for large length of input and output parameters this method is not applicable anymore. Authors determined the Walsh spectrum for Boolean functions with input length of maximum 16 bits, but the resulting storing file for this amount of data was as big as 128MB. They concluded that above this input length, the Walsh spectrum method cannot be used anymore.

As a consequence, the authors tried to avoid the limitations imposed by the Walsh spectrum method. For this reason they considered determining the input and output corresponding linear expressions of a much simpler Boolean function using the method of solving linear systems of equation. The Boolean function going to be analyzed is defined as:

$$f : \mathrm{GF}(2^8) \rightarrow \mathrm{GF}(2^8) \quad (1)$$

$$f(x) = 32x \oplus x \quad (2)$$

If the result of $f(x)$ exceeds 8 bits in length, then it is truncated to 8 bits in order to remain in the dimension of $\mathrm{GF}(2^8)$. Only 8 LSB (Least Significant Bits) are kept, the rest of the bits being ignored. It is beyond any doubt that $f(x)$ is a linear Boolean function.

The function $f(x)$ was studied first using the Walsh spectrum method. From the Walsh spectrum, the authors selected some input / output linear expressions according to Table 1. Due to the fact that the function $f(x)$ is linear, the probability for all input and output linear expressions equals 1.

Table 1. Input / output linear expressions for $f(x)$ (selection).

Input linear expression	00001011	01000101	01111111	10101110
Output linear expression	00001011	01000111	01111100	10101011
Probability	1	1	1	1

The selected pairs of input and output linear expressions are used to build a linear system of equations, as follows:

(1) Eight values of input data are selected at random ($x_1 \ldots x_8$);
(2) The corresponding values of output data are calculated ($f(x_1) \ldots f(x_8)$);
(3) Input data is independently masked (BitAnd) with the value of the input linear expression;
(4) Even bit parity for every result in step (3) is independently calculated;
(5) The binary representation for output values $f(x_1)$ is used as coefficients for the binary variables which form the output linear expression to be determined.

The authors chose two sets of input data that are detailed in Table 2 and Table 3. The reason for choosing two sets of input data is that for the first set of data the value of the system determinant was null. As a consequence, the linear system solution is not singular. To be noted that the system of equations is solved in binary fields. In order to do calculations in binary fields, the authors developed and used some programs in Wolfram Mathematica ® to solve the linear systems of equations described below. These programs were used in conjunction with a third party Galois Field Package developed for Wolfram Mathematica ® [6].

Table 2. Input/ output set of data used to determine the linear expression of $f(x)$ (set 1).

Input value x (decimal)	160	127	222	122	82	92	38	77
Output value f(x) (decimal)	160	159	30	58	18	220	230	237
Determinant (binary)	0							

Table 3. Input/ output set of data used to determine the linear expression of $f(x)$ (set 2).

Input value x (decimal)	199	127	222	122	149	71	38	77
Output value f(x) (decimal)	39	159	30	58	53	167	230	237
Determinant (binary)	1							

For the first set of data (Table 2), a linear system of equations with coefficients in binary fields can be constructed as follows:

$$\begin{cases} x_1 + x_3 = 0 \\ x_1 + x_4 + x_5 + x_6 + x_7 + x_8 = 1 \\ + x_4 + x_5 + x_6 + x_7 = 0 \\ + x_3 + x_4 + x_5 + x_7 = 0 \\ + x_4 + x_7 = 1 \\ x_1 + x_2 + x_5 + x_6 = 1 \\ x_1 + x_2 + x_3 + x_6 + x_7 = 1 \\ x_1 + x_2 + x_3 + x_5 + x_6 + x_8 = 0 \end{cases} \quad (3)$$

Upon solving the linear system (3), the solution for output linear expression is not as expected in Table 1. Instead of getting the solution $[x_1 \ldots x_8] = [00001011]$, we found $[x_1 \ldots x_8] = [11100110]$, due to the fact that the solution is not singular. However, after building up the same system with input / output values from Table 3, the authors found the solutions exactly as expected, which confirms the method.

In the process of solving linear systems with binary fields coefficients, the authors encountered some issues worth mentioning:

(1) If the determinant of the system is null, the solution of the system ($x_1 \ldots x_8$) is not singular. In this case, changing the random set of input data will solve the problem;

(2) If the determinant of the system is null or not null, however the solution of the system ($x_1 \ldots x_8$) cannot be found, and the message "*Power: GF2[0]^-1 and GF2[0]^0 are undefined*" is showed, this situation is due to the fact that the coefficient of x_1 in first equation is null, and switching the equations accordingly will solve the problem.

This method of determining input/output linear expression for linear functions is very useful in the cases where the Walsh spectrum method is not applicable. The linear systems method was successfully tested by the authors for larger input/output bit-length (32 bit) and they concluded that the method is reliable and is practically independent from a certain bit-length. It worth mentioning that for input/output bit-length larger than 16 bits, the determinant of the linear system cannot be calculated in real time and increasing latency is expected. The authors experimented with different dimensions for input/output bit-length and noticed a latency of 45 minutes for 32 bits, compared with a latency of few seconds for 16 bits. However, the linear system was solved in few seconds for 32 bits, compared to almost instantaneously for 16 bits. The authors concluded that for larger input/output bit-length, it is only possible to solve the system "in blind" without knowing the value of the determinant.

In order to evaluate how appropriate the method of solving the linear systems for determining linear expression for non-linear functions is, the authors tried to run the method described above for the AES S- box. A linear system of equations was built, but the corresponding values of output data ($f(x_1) \ldots f(x_8)$) were calculated using the AES S-box look-up table. For the set of data given in Table 4, the corresponding linear system of equations is given in (4).

Comparing input/output linear expression obtained from the linear systems method with the expected input/output linear expression obtained from the Walsh spectrum method, the authors concluded that these values are among the other 239 possible linear expressions. However, the method can't distinguish the high linear probability expressions. Analyzing AES S-box through the Walsh spectrum method, the authors determined that for this input/output linear expression the value of linear probability is $(6/64)^2$ and the bias is $\varepsilon = 2^{-4.41}$.

$$\begin{cases} x_1 + x_3 = 1 \\ x_2 + x_5 + x_6 + x_7 + x_8 = 0 \\ x_3 + x_4 = 1 \\ x_2 + x_4 + x_5 = 1 \\ x_4 + x_5 = 0 \\ x_1 + x_2 + x_4 + x_5 + x_6 = 0 \\ x_1 + x_2 + x_3 + x_6 + x_7 = 0 \\ x_1 + x_2 + x_3 + x_5 + x_6 + x_8 = 0 \end{cases} \quad (4)$$

Table 4. Input / output set of data used to determine the linear expression of *AES S-Box*

Input value x (decimal)	71	146	8	94	52	147	245	83
Output value f(x) (decimal)	160	79	48	88	24	220	230	237
Determinant (binary)	1							
Input linear expression(decimal)	58							
Output linear expression (decimal)	103							
Bias	$2^{-4.41}$							

4. Conclusions

The authors tried to test the applicability and usefulness of two methods (the Walsh spectrum and linear systems) on finding linear expressions of linear and nonlinear Boolean functions and came to the following conclusions:

- For linear Boolean functions both methods are applicable and reliable, but overall these functions are large in input bit-length (over 32 bits). Therefore, the linear systems method is preferable.
- For nonlinear Boolean functions only the Walsh spectrum method is reliable, because these functions are usually small in input bit-length (under 8 bits).

5. References

[1] LUCIA LACKO-BARTOSOVA, *Linear and differential cryptanalysis of reduced-round AES*, Tatra Mountains Mathematical Publications, pp. 51-61, 2011

[2] JOAN DAEMEN, VINCENT RIJMEN, *The Design of Rijndael AES - The Advanced Encryption Standard*, Springer, 2002

[3] MITSURU MATSUI, *Linear cryptanalysis method for DES cipher*, Advances in Cryptology – EUROCRYPT '93, Workshop on the Theory and Application of Cryptographic Techniques, Lofthus, Norway, 1993 (T. Helleseth, ed.), Lecture Notes in Comput. Sci., Vol. 765 Springer-Verlag, Berlin, 1994, pp. 386–397

[4] MITSURU MATSUI, *The first experimental cryptanalysis of the Data Encryption Standard*, Advances in Cryptology – CRYPTO '94, 14th Annual Internat. Cryptology Conf., Santa Barbara, CA, USA, 1994 (Y. G. Desmedt, ed.), Lecture Notes in Comput. Sci., Vol. 839, Springer-Verlag, Berlin, 1994, pp. 1–11

[5] HOWARD HEYS, *A tutorial on linear and differential cryptanalysis*, Technical Report CORR 2001-17, Centre for Applied Cryptographic Research, Department of Combinatorics and Optimization, University of Waterloo, March 2001

[6] RYOH FUJI-HARA, *Galois Field Package Manual*, https://infoshako.sk.tsukuba.ac.jp/jcca/GaloisField/Doc/GF- PackageManual(E).pdf

[7] NIST, *Specification for the ADVANCED ENCRYPTION STANDARD (AES)*, Federal Information Processing Standards Publication 197, 2001

[8] LIU JINGMEI, WEI BAODIAN, WANG XINMEI, *New Method to Determine Algebraic Expression of Rijndael S-box*, InfoSecu04, pp. 181-185, 2004

[9] SHRISTI DEVA SINHA, CHAMAN PRAKASH ARYA, *Algebraic Construction and Cryptographic Properties of Rijndael Substitution Box*, Defence Science Journal, Vol. 62, No. 1, pp. 32-37, January 2012