BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 62 (66), Numărul 3, 2016
Secția
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

CONSTANT-WEIGHT SUBSTITUTION CODES AGAINST SIDE-CHANNEL ATTACKS

BY

LUMINIȚA SCRIPCARIU*

Technical University “Gheorghe Asachi” of Iaşi,
Faculty of Electronics,Telecommunications and Information Technology

Received: August 25, 2016
Accepted for publication: September 30, 2016

Abstract. Side-channel attacks represent serious threats for any security system. Their success is based on analysing the implementation of the cryptographic algorithms and finding some weaknesses of the hardware or software cryptosystem. One major cryptographic weakness is given by the variable time or energy used to process different numerical values, depending on their binary representation and its Hamming weight. If all the numeric values used by the cryptographic algorithm would have the same Hamming weight, than the processing time and the power consumption would be the same for all values and the attacker could not deduce any information by measuring it. We propose, describe and analyze some substitution codes which generate binary sequences with the same Hamming weight and can be used as countermeasures against side-channel attacks. Finally, an example of how to implement a CWS code in software with an S-box and in hardware as a logical circuit is detailed.

Key words: cryptography; side-channel attack; codes; Hamming weight.

1. Introduction

Information security is the major goal of any cryptosystem. Different cryptography codes are used in order to secure data transmitted by various communication channels (Paar & Pelzl, 2010). Public keys or secret keys

*Corresponding author: *e-mail*: lscripca@etti.tuiasi.ro

encryption algorithms processes input values using specific hardware implementations.

Side-channel attacks are powerful hacking methods which deduce the encryption keys or some information values based on the algorithm processing time (time attacks) or on the analysis of the power consumption of the target device or on other aspects (Black & Urtubia, 2002).

Simple power analysis (SPA) is a side-channel attack used on RSA algorithm (Mangard, 2002).

Differential Power Analysis (DPA), based on the analysis of the variations of the electrical power consumption of a cryptographic module, is another side-channel attack successfully used against DES and TDES (McEvoy *et al.*, 2007).

Binary Power Analysis (BPA) is a variant of DPA used to break AES algorithm.

Comparative Power Analysis (CPA), made as a self-similarity attack, can be applied to RSA (Homma *et al.*, 2010).

The implementation of any encryption algorithm can be done in various modes (Ferguson *et al.*, 2010). Attackers exploit the weaknesses of these hardware or software implementations so it is important to test them very seriously before using them.

The processing time of the algorithm and the power consumption of the cryptographic device depend on the Hamming weight of the binary form of a numerical value. If the Hamming weight of the input value set is uniformed using a substitution code, than this kind of information is no longer available to the attackers.

In this paper, an analysis of possible substitution codes is made and some recommendations about optimal dimensions of these codes are given in the end.

The proposed code type is presented in the second paragraph. Some convenient codes are selected and compared in the third section of the paper. Some principles used to design these codes are described in the fourth paragraph. Finally some conclusions are presented followed by the references used as documentation.

2. About Constant-Weight Substitution Codes

The input values of any encoder have usually variable Hamming weight which is expressed as the number of 1's in the binary sequence associated to a value.

If we want to increase the robustness of a cryptosystem against side-channel attacks and to reduce the number of arithmetical operations made by an algorithm, constraints regarding the Hamming weight of the input value must be imposed. Specific channel codes have to be applied on the input values in order to satisfy these constraints.

These codes are intended to maintain the same Hamming weight for all the code words. Therefore we call them Constant-Weight Substitution (CWS) codes.

Let us consider that the input values can be expressed on *N* bits.

The output alphabet of the code can be expressed on *M* bits and the Hamming weight of each code value is equal to *k*. The number of possible *M*-bit sequences with the weight equal to k is expressed combinatorically as:

$$m = \binom{M}{k}. \tag{1}$$

Each input value of N bits is substituted by an output *M*-bit sequence containing k 1's based on a substitution table called S-box.

The code is described by the combination of three parameters *M, k* and *N* and it is denoted as CWS (*M, k, N*).

It is necessary to satisfy the following inequality:

$$2^N \le \binom{M}{k}. \tag{2}$$

So the maximum number of input bits is equal to:

$$N_{\max} = \left\lfloor \log_2 \binom{M}{k} \right\rfloor. \tag{3}$$

The coding rate is expressed as percentages using the relation below:

$$R = \frac{N}{M} \cdot 100(\%) \tag{4}$$

Different combinations of values (*M, k,* $N_{\max}$) and the coding rate of the corresponding CWS code are presented in Tables 1 and 2.

There are presented only those dimensions which create a code with at least two input bits and a coding rate greater than 50%.

Small Hamming weight values (*k*) are preferred even if:

$$\binom{M}{k} = \binom{M}{M-k} \tag{5}$$

The proposed encoding method aims to reduce the Hamming weight so only the minimum value of the pair (*k, M – k*) is written in this table.

Some codes have the Hamming weight equal to a half of the coding length or 50 % of the bits are equal to ‘1’. These codes have an equal number of ‘1’s and ‘0’s so they are balanced codes.

3. Analysis of Constant-Weight Substitution Codes

Different codes can be chosen for a specific transmission or encryption algorithm. Therefore some tables with possible combinations of code parameters are useful to make such decision.

Tables 1 and 2 the parameters of some "very small" and “small” codes, for which the input sequence length (N) is at most 8 or 16, respectively, while Table 3 presents "large" CWS codes, with large values of M and N, up to 64.

Table 1
Parameters of some "very small" CWS codes

Code word length (M)	Hamming weight (k)	No. of code sequences	Maximum input sequence length (N)	Coding rate (R)
4	1	4	2	50%
4	2	6	2	50%
5	2	10	3	60%
6	2	15	3	50%
6	3	20	4	66%
7	2	21	4	57%
7	3	35	5	71%
8	2	28	4	50%
8	3	56	5	62%
8	4	70	6	75%
9	2	36	5	55%
9	3	84	6	66%
10	2	45	5	50%
10	3	120	6	60%
10	4	210	7	70%
11	3	165	7	63%
11	4	330	8	72%
12	2	66	6	50%
12	3	220	7	57%
12	4	495	8	66%
13	3	286	8	61%
14	3	364	8	57%
15	3	455	8	53%

Usually the coding rate must be as great as possible while it is an advantage to have a low Hamming weight in order to reduce the number of operations made by an encoding algorithm.

The “very small” CWS codes can have a maximum coding rate of 75% while the “small” ones can reach 84%.

Table 2
Parameters of Some "Small" CWS Codes

Code word length (*M*)	Hamming weight (*k*)	No. of code sequences	Maximum input sequence length (*N*)	Coding rate (*R*)
13	4	715	9	69%
13	5	1,287	10	76%
14	4	1,001	9	64%
14	5	2,002	10	71%
14	6	3,003	11	78%
15	4	1,365	10	66%
15	5	3,003	11	73%
15	6	5,005	12	80%
16	3	560	9	56%
16	4	1,820	10	62%
16	5	4,368	12	75%
16	7	11,440	13	81%
17	4	2,380	11	65%
17	5	6,188	12	70%
17	6	12,376	13	76%
17	7	19,448	14	82%
18	3	816	9	50%
18	4	3,060	11	61%
18	5	8,568	13	72%
18	6	18,564	14	77%
18	8	43,758	15	83%
19	5	11,628	13	68%
19	6	27,132	14	73%
19	7	50,388	15	79%
19	8	75,582	16	84%
20	3	1,140	10	50%
20	4	4,845	12	60%
20	5	15,504	13	65%
20	6	39,760	15	75%
20	7	77,520	16	80%
21	5	20,349	14	67%
21	6	54,264	15	71%
21	7	116,280	16	76%
22	4	7,315	12	55%
22	5	26,334	14	63%
22	6	74,613	16	73%
23	5	33,649	15	65%
23	6	100,947	16	69%
24	4	17%	13	54%
25	4	16%	13	52%
26	4	15%	15	57%

Table 3
Parameters of Some "Large" CWS Codes

Code word length (*M*)	Hamming weight		Maximum input sequence length (*N*)	Coding rate (*R*)
	k	%		
22	7	32%	17	77%
22	8	36%	18	82%
22	10	45%	19	86%
23	7	30%	17	74%
23	8	35%	18	78%
23	9	39%	19	82%
23	10	43%	20	87%
24	6	25%	17	71%
24	12	50%	21	87%
25	12	48%	22	88%
26	13	50%	23	88%
27	12	44%	24	89%
28	11	39%	24	86%
32	5	15%	17	53%
32	8	25%	23	71%
32	9	28%	24	75%
32	10	31%	25	78%
32	11	34%	26	81%
32	12	37%	27	84%
32	13	40%	28	87%
32	15	46%	29	90%
64	**8**	**12.5%**	**32**	**50%**
64	12	18%	41	64%
64	16	25%	48	75%
64	23	36%	56	87%
64	**32**	**50%**	**60**	**94%**

Analyzing the third table, we notice that the large CWS codes can reach a coding rate of 94% which is recommended for high-speed communication systems.

Large code sizes multiple of 8 can be expressed as a number of bytes (B) in order to be easier to write the code table.

For example, CWS (64, 23, 56) code is applied on 7-byte sequences and generates 8-byte sequences so the coding rate is 7B:8B.

CWS (64, 8, 32) code is a minimum weight code, with a coding rate of 4B:8B.

CWS (32, 9, 24) code has 3-byte sequences as input and 4-byte sequences as output so the coding rate is 3B:4B.

These tables can be used to choose a proper code for a specific application.

For example, five different codes can be used for 8-bit input sequences: CWS (11, 4, 8), CWS (12, 4, 8), CWS (13, 3, 8), CWS (14, 3, 8) or CWS (15, 3, 8).

The criterion used to choose the proper code can be the maximum coding rate or the minimum Hamming weight, expressed as a percentage value.

A compromise between these two criteria is another option in order to make a decision.

In this particular case, we can choose CWS (15, 3, 8) code having the minimum Hamming weight. This code transmits a low number of '1's, about 20% of the total number of transmitted bits.

We can also choose CWS (11, 4, 8) with the maximum coding rate if a high transmission speed is our priority.

CWS (13, 3, 8) code is the best choice if a compromise between the two criteria is made.

Comparing the codes from these three tables, the minimum Hamming weight (12.5%) corresponds to the extra-large code CWS (64, 8, 32) with a coding rate of 4B:8B or 50%.

The maximum coding rate (94%) corresponds to CWS (64, 32, 60), a large code which can be designed using hexadecimal numbers. This code can be used for high-speed communication processes which require low redundancy.

These tables specify the maximum number of input bits but CWS codes can be designed for a more convenient number of bits less than the maximum value.

For example, CWS (24, 6, 16) code can be designed using 16 input bits instead of 17, as it is mentioned in Table 3 for 24-bit code sequences with the Hamming weight equal to 6. It has a coding rate of 2B:3B and the design is simplified by the fact that the input and output length s are multiple of 8.

4. Principles of Constant-Weight Substitution Codes Design

"Small" codes are designed and implemented easier than "large" codes which works with large numbers and very large substitution tables.

Having small sizes, these CWS codes can be implemented on hardware, with logical circuits, or as software modules, based on the pre-calculated S-boxes. It is easy to calculate the code table and store it in the system memory, in order to minimize the coding time of the encryption algorithm.

One way to simplify the design process of a CWS code is using hexadecimal numbers. This way can be followed when the sequence length is a multiple of 4.

Let us consider the CWS (8, 2, 4) code which is a small-size low-density code.

Only the 8-bit sequences with a Hamming weight of 2 can be used as code sequences. There are 28 sequences of this kind. The code uses only 16 of them. The S-box of this code with hexadecimal output values is presented in Table 4. The first two bits of the input sequence specify the row and the last two bits of it specify the column of the table. For example, the decimal input value 12 (binary sequence: 1100) is substituted by the hexadecimal value 81 corresponding to the decimal value 129.

Table 4
S-Box of CWS (8, 2, 4) Code

Input	00	01	10	11
00	11	12	14	18
01	21	22	24	28
10	41	42	44	48
11	81	82	84	88

Let us denote the binary input and output sequences as:

$$x = \overline{x_3 x_2 x_1 x_0}\,, \mathrm{y} = \overline{y_7 y_6 y_5 y_4\, y_3 y_2 y_1 y_0} \tag{6}$$

This code can be implemented with a hardware structure described by the following logical relations:

$$y_0 = (x_0 + x_1)',\ y_1 = (x'_0 + x_1)', y_2 = (x_0 + x'_1)', y_3 = (x'_0 + x'_1)' \tag{7}$$

$$y_4 = (x_2 + x_3)', y_5 = (x'_2 + x_3)', y_6 = (x_2 + x'_3)', y_7 = (x'_2 + x'_3)' \tag{8}$$

For large CWS codes, it is really difficult to write the code table or the logical circuit equations. A large amount of memory must be used to store the code values. To design a large CWS code, it is an advantage to work directly on bytes instead of bits or using hexadecimal numbers to compact the S-box. This principle can be applied for the codes with the input and output lengths equal to multiples of 4 or 8.

The design process can be also simplified and the S-box sizes can be reduced if some symmetry of the code is noticed and exploited.

Other numerical bases can be used to reduce the CWS codes design complexity.

5. Conclusions

Many encryption algorithms are vulnerable to side-channel attacks. the proposed constant-weight substitution (cws) codes can precede any encryption module in order to reduce the calculus complexity, to uniform the processing time and energy and to counter fight the side-channel attacks. different cws codes are presented, up to 64-bit code sequences. some principles of designing this kind of codes are described. the proposed cws codes can be implemented in software with s-boxes and in hardware with logical circuits. different numerical bases can be used in order to compact the values used by these codes and to simplify the design of them.

REFERENCES

Black J., Urtubia H., *Side-Channel Attacks on Symmetric Encryption Schemes: The Case for Authenticated Encryption*, Proc. of the 11th USENIX Security Symposium, San Francisco, USA, 327-338, 2002.

Ferguson N., Schneier B., Kohno T., *Cryptography Engineering – Design Principles and Practical Applications*, Wiley Publishing, Inc., 2010.

Homma N., Miyamoto A., Aoki T., Satoh A., Shamir A., *Comparative Power Analysis of Modular Exponentiation Algorithms*, IEEE Trans. Comput., **59**, *6*, 795-807 (2010).

Mangard S., *A Simple Power-Analysis (SPA) Attack on Implementation of the AES Key Expansion*, Proc. of Inform. Security and Cryptology, ICISC 2002, 343-358, 2002.

McEvoy R., Tunstall M., Murphy C.C., Marnane W.P., *Differential Power Analysis of HMAC Based on SHA-2, and Countermeasures*, 8th Workshop on Inform. Security Applications – WISA 2007, Lecture Notes in Computer Science, **4867**, 317-332 (2007).

Paar C., Pelzl J., *Understanding Cryptography*, Springer-Verlag Berlin, 2010.

CODURI DE SUBSTITUȚIE CU PONDERE CONSTANTĂ UTILIZATE ÎMPOTRIVA ATACURILOR PE CANALE LATERALE

(Rezumat)

Atacurile pe canale laterale sunt o amenințare la adresa securității sistemelor informatice bazate pe analiza implementării algoritmilor criptografici. Un punct slab al acestora îl constituie variabilitatea timpului şi energiei folosite pentru procesare, dependente de forma binară a numerelor şi de ponderea lor Hamming. Dacă toate valorile folosite de algoritmul de criptare ar avea aceeaşi pondere, atunci timpul de procesare şi energia necesară ar fi aceleaşi iar atacatorul nu ar putea obține informații prin măsurarea lor. În lucrare, sunt propuse, descrise şi analizate coduri de substituție cu pondere constantă care pot fi folosite pentru contracararea atacurilor pe canale laterale.