Possibilities and testing of CPRNG in block cipher mode of operation PM-DC-LM

Petr Zacek, Roman Jasek, and David Malanik

Citation: AIP Conference Proceedings **1738**, 120029 (2016); doi: 10.1063/1.4951912
View online: http://dx.doi.org/10.1063/1.4951912
View Table of Contents: http://aip.scitation.org/toc/apc/1738/1
Published by the American Institute of Physics

Possibilities and Testing of CPRNG in Block Cipher Mode of Operation PM-DC-LM

Petr Zacek, Roman Jasek and David Malanik

Faculty of applied Informatics, Tomas Bata University in Zlin, Zlin, Czech Republic
{zacek, jasek, dmalanik}@fai.utb.cz

Abstract. This paper discusses the chaotic pseudo-random number generator (CPRNG), which is used in block cipher mode of operation called PM-DC-LM. PM-DC-LM is one of possible subversions of general PM mode. In this paper is not discussed the design of PM-DC-LM, but only CPRNG as a part of it because designing is written in other papers. Possibilities, how to change or to improve CPRNG are mentioned. The final part is devoted for a little testing of CPRNG and some testing data are shown.

Keywords: Deterministic Chaos, Logistic Map, CPRNG, Symmetric Cryptography, Block Cipher, Block Cipher Mode of Operation, PM-DC-LM
PACS: 05.45.Gg

INTRODUCTION

Our designed polymorphous mode PM with its subversion PM-DC-LM is described in material [6] even though there has not a name. This paper is not about designing, but about testing the one part of PM-DC-LM, CPRNG.

Motivation of this pare is to test the behavior of used CPRNG built on deterministic chaos – logistic maps and discuss possibilities about used CPRNG.

Possibilities about deriving initialization values for CPRNG from *IV* (initialization vector) are discussed. The last part is about testing of CPRNG on real data. Real data will be two random generated *IV* 256-bit length. The value x_n will be calculated one million times.

DESCRIPTION OF PM-DC-LM MODE

The acronyms "PM", "DC", and "LM" mean it is "Polymorphous Mode" with using the "Deterministic Chaos", and "Logistic Maps" were used as type of deterministic chaos. This name is variable for the future purposes and PM is basis. The mode is mostly described in [2, 4], where it is designed for the first time. But some changes were done. This mode has not fixed structure, however, we changed the equation of calculation for the next key. Contrast to the original, there is used one more value *g*. Value *g* is derived as the last three digits of value x_n from CPRGN.

DESCRIPTION OF THE USED CPRNG

This part is about description of used CPRNG. The CPRNG is fully described and it is same as in material [6]. But briefly, the CPRNG is based on the deterministic chaos logistic map, which operates by following equation.

$$x_n = r x_{n-1}(1 - x_{n-1}) \tag{1}$$

The *r* is the control parameter, and x_n is the actual value and x_{n-1} is previous generated value. If the *r* is a real number above 3.57, the system behaves chaotically for almost values[1]. On this fact, we choose real numbers on the interval <3.9, 4.0) to avoid numbers near the 3.82842712... In this generator, the number *r* is generated from *IV*. Value x_i should be a real number on the interval (0, 1). The first value $x \rightarrow x_0$ will be calculated by following equations, where the first is for values $r > 3.9$ and the second is for $r = 3.9$. [1, 3, 5]

[1] For the r = 3.82842712... the system oscillates among three values – sometimes i tis called „islands of stability" [2]

International Conference of Numerical Analysis and Applied Mathematics 2015 (ICNAAM 2015)
AIP Conf. Proc. 1738, 120029-1–120029-4; doi: 10.1063/1.4951912
Published by AIP Publishing. 978-0-7354-1392-4/$30.00

$$x_0 = 10(r - 3.9) \quad (2)$$

$$x_0 = 10^{-15} \quad (3)$$

The value r is calculated from initialization vector before encryption. The algorithm how to calculate value from initialization vector is also described in material [6].

PROPERTIES OF PROPOSED CPRNG AND POSSIBILITIES

Because of the polymorphous structure of proposed mode the parts could be changed. According to CPRNG we can change it as following.

- We can change the type of deterministic chaos
- We can change principle, how to calculate values *r* and x_0 as their precision

Type of Deterministic Chaos

As the other parts could be changed also the type of deterministic chaos could be changed. As well as the type the principle, how to derive values for CPRNG, must be changed.

Derivation of Values *r* and x_0

The actual derivation of value *r* is following and it also described in material [6].

1. Express *IV* as binary number.
2. Split the *IV* into same blocks with length as precision *p*.
3. Represent it as numbers 0 and 1.
4. Count the sum of numbers 0 and 1 at their corresponding position modulo 10.
5. Concatenate sums as digits after to 3.9.

Ideally we want to have the different CPRNG for all IVs. Above principle is not the best. It leads to collisions. For example, if we have two *IV*s IV_1 = 0110 and IV_2 =1001, the value *r* will be the same -> *r* = 3.91111. There are two ways, how to avoid mentioned collisions.

1. Use the non-collision hash function before converting the *IV* into value *r*.
2. Representing the *IV* as decimal number with fixed length (including leading zeros) and then concatenate all digits after to value *r*.

Above possibilities have advantages and disadvantages as well. The first possibility should be secure, if we use non-collision function, but it will not avoid them at all. It will be also quicker and we will not have problems with representation of long float numbers. The second possibility will not lead to collisions at all, but we will need to operate with long float numbers. For example the value r computed from *IV* 256-bit length will have 79 digits after decimal point and it could be problem.

Derivation of value x_0 could be changed as well.

- We can derivate it independently to value *r*.
  - Use another different algorithm from derivation algorithm of value *r* for same *IV*
  - Use the second *IV* for value x_0

TESTING OF CPRNG

Values *d* and *g*

As the first we tested generating of the first ten values from two different generator of deterministic chaos based on two randomly generated different *IV*s with length 256 bits, which are different in the last bit. The Precision *p*

(number of digits in value r after 3.9) of the r was chosen 14 (maximal in Python 3.x which was used for testing). Here they are results.

IV_1 = 111111…1000
IV_2 = 111111…1001

r_1 from IV_1 = 3.911132271809247
r_2 from IV_2 = 3.911132271809248

x_1 = 0.9402137244001353
x_2 = 0.94021372440381

First ten values from generator number one based on r_1 and x_1
[0.3869282003850097, 0.9277783349901382, 0.26206814046285243, 0.7563677304170022, 0.720726194443333, 0.787232496718318, 0.6551048496504891, 0.8836909470612287, 0.40199109175011855, 0.9402137244001353]

First ten values from generator number two based on r_2 and x_2
[0.38692820038504017, 0.9277783349901654, 0.2620681404627615, 0.756367730416833, 0.7207261944436724, 0.7872324967177322, 0.6551048496518054, 0.8836909470596319, 0.40199109175491127, 0.94021372440381]

First ten values g from the generator number one and corresponding first ten values d (highlighted by red color)
[097,382,243,022,333,318,891,287,855,353]
First ten values g from the generator number two and corresponding first ten values d (highlighted by red color)
[017,654,615,833,724,322,054,319,127,381]

As we can see from results above, the first ten values g from generator number two are at 100% different from the first ten values g from generator number one. The value d is different in 90%. So, although the IVs are different in only one bit, the generators will behave totally different.

Ideal frequency of values d and g should be indistinguishable from frequency of random. For d probability should be 1/9 for any number from 1 to 9. Because of using the deterministic chaos it will be distinguishable from random behavior. For demonstration we tested CPRNG for the first 1000000 values x_n. The results are shown in Figures 1-2.

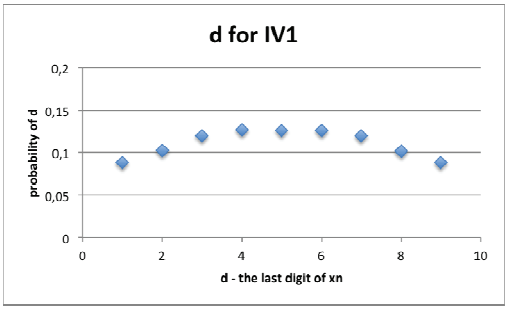

FIGURE 1. Probabilities for d of IV1

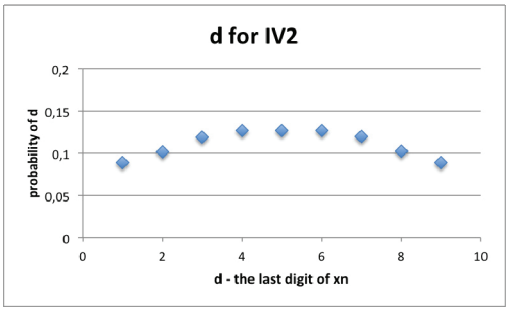

FIGURE 2. Probabilities for d of IV1

As we can see, the probabilities are distinguishable from random-looking. For more random-looking should be used another type of pseudo-random number generator.

CONCLUSION

In this paper was tested CPRNG based on logistic maps, which is used in our mode PM-DC-LM. There were discussed possibilities and ways, how to upgrade or change the CPRNG. There was tested one of the possible ways,

how to set CPRNG on two random generated *IV*s. The CPRNG was run one million times and probabilities of values *g* and *d* were shown in graphs.

It is obvious CPRNG is distinguishable from random-looking, because it is deterministic, but whole distribution looks good and stable.

For future research or upgrade would be good to try to change type of deterministic chaos or try to change the CPRNG to PRNG. It would be also interesting to change algorithm, how the values r and x_n are derived from *IV*.

ACKNOWLEDGMENTS

This work was supported by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2015/47; further it was supported by financial support of research project NPU I No. MSMT-7778/2014 by the Ministry of Education of the Czech Republic; also by the European Regional Development Fund under the Project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

REFERENCES

1. Bellovin, S. M. Columbia University. Modes of Operation [online]. Columbia, USA, 2009 [cit. 30.1.2015]. Online at: https://www.cs.columbia.edu/~smb/classes/s09/l05.pdf
2. Weisstein, Eric W. "*Logistic Map*." From MathWorld-A Wolfram Web Resource. http://mathworld.wolfram.com/LogisticMap.html
3. Modes Development. In: National Institute of Standards and Technology: Computer Security Resource Center [online]. 2001, 16.10.2014 [cit. 2015-02-03]. Online at: http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html
4. Sprott, J. C., Chaos and Time-Series Analysis, Oxford University Press, 2003
5. Senkerik R, Pluhacek M, Zelinka I, Davendra D, Oplatkova Z. A Brief Survey on the Chaotic Systems as the Pseudo Random Number Generators. In: Sanayei A, E. Rössler O, Zelinka I (eds) ISCS 2014: Interdisciplinary Symposium on Complex Systems, vol 14. Emergence, Complexity and Computation. Springer International Publishing, pp 205-214. doi:10.1007/978-3-319-10759-2_22, 2015.
6. Zacek, P., R. Jasek, and D. Malanik, Using the Deterministic Chaos in Variable Mode of Operation of Block Ciphers, in Artificial Intelligence Perspectives and Applications, R. Silhavy, et al., Editors. 2015, Springer International Publishing. p. 347-354, DOI: 10.1007/978-3-319-18476-0_34