A Comparison of the PM-DC-LM Mode With Other Common Operational Block Cipher Modes

Petr Zacek, Roman Jasek, David Malanik
Faculty of Applied Informatics
Tomas Bata University
Zlin, Czech Republic
e-mail: {zacek, jasek, dmalanik}@fai.utb.cz

***Abstract* — The aim of this paper is to compare the performance of Polymorphous Mode - Deterministic Chaos - Logistic Maps (further only PM-DC-LM) with some of the most commonly used block cipher modes of operation. Among the most notable of these are the Electronic Code Book (ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR). In order to do so, the exclusive OR (XOR) function - instead of a regular block cipher, was used as the encryption algorithm with a Bitmap (BMP) image being used as input data. The testing codes were written in the Python Version 3.4 programming language. Additionally, various operational modes were compared against each other using basic properties like speed or error propagation, among others. The results indicate that PM-DC-LM would seem to be more "random-looking" than the other modes it was tested against and, although the XOR function was used, the output data looked as though it was encrypted. The results also demonstrate the possible advantages in using a polymorphous structure on an image file (i.e. repetitive blocks of data) when compared with the other modes even where the non-standard conditions were set.**

Keywords - cryptography; block cipher; mode of operation; PM-DC-LM; polymorphism; deterministic chaos; logistic map.

I. Introduction

Most of the generally approved block cipher operational modes are quite old [2]. This is not a disadvantage, but an attempt has been made to do things differently. There are five commonly used operational modes – namely, the ECB mode, the CBC mode, the CFB mode, the OFB mode, and the CTR mode; as described in [2]. All of these block cipher modes provide certain advantages as well as disadvantages, and therefore, their appropriate use or application should always be taken into consideration. For example, the ECB mode was found to be really quick - but not secure, when compared to the other modes; while the CTR mode does show proof of security [8]. These five modes are generally used in order to ensure the confidentiality of encrypted systems. There have been prior attempts at designing new modes [3], however none of them are polymorphous in structure, thereby not offering the chance to explore the PM-DC-LM mode. Moreover, only one mode was designed with the principle of changing the key – i.e. the Key Feedback Mode [7].

Additionally, all of these modes are considered as being fixed to their structure, and are often used as a complement to a block cipher. In view of this, the design of a new group - - called the Polymorphous Mode - Deterministic Chaos - Logistic Map (PM-DC-LM) was explored This mode is derived for example from the Polymorphous Mode (PM); which is not an entirely new proposal. The following section presents a brief introduction of this group (mode) and a brief introduction of some changes that were tested in this paper.

This paper compares the PM-DC-LM mode - as one possible example derived from PM, with the other five previously mentioned modes. The comparison was based on the speed, image data, and other properties of block cipher modes (e.g. parallelizability, error propagation, how a key I affected, security, etc.). This paper – as compared to [1], has been drafted to show the preliminary results of the initial research.

All of these modes were written and tested in the Python, Version 3.4, programming language. The XOR function was used for the "encryption algorithm". It is a well known fact that security is based on the block cipher algorithm. However, attempts were made to test it using the XOR function ("without approved block cipher algorithm").

Section 2 provides a brief description of the PM and PM-DC-LM modes. Section 3 is concerned with the fundamental issues and an introductory comparison. In Section 4, the results arising from testing the PM-DC-LM mode on the base of image data are shown. Section 5 presents the results derived from speed comparisons. In Section 6, the modes are compared on the base of other properties.

II. Introduction to (PM) - PM-DC-LM

PM-DC-LM is the acronym for Polymorphous Mode - Deterministic Chaos - Logistic Map. The Polymorphous Mode means that this mode makes variable use of previous plain text, cipher text, and a key to calculate the key for the encryption of the next block of plain text. Deterministic Chaos Mode (DM), on the other hand, is used to determine how the plain text, cipher text and key are used to calculate the key. The Logistic Map represents the type of deterministic chaos.

The PM-DC-LM mode is one possible mode that can be derived from the PM mode. The PM mode represents the main idea about polymorphous modes; therefore, all other modes are fixed to their main structure. This means that they are straightforward and they have only one "way". Our efforts were directed to making a group of modes whose structure is determined by Deterministic Chaos.

ISBN: 978-1-61208-427-5

This mode can be adapted in many ways because one can change the adjustment of the Chaotic Pseudo-Random Number Generator (CPRNG); or replace Deterministic Chaos with another Pseudo-Random number generator or the function for calculating the new key. This mode is derived from another paper [1], where the mode is described in greater detail. For testing purposes, the chosen modes were almost similar, and adjusted as follows:

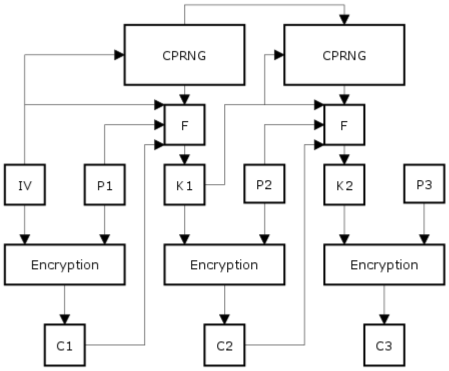


Figure 1. Diagram of PM-DC-LM mode

IV – Initialization vector
C – Cipher text
P – Plain text
K – Key
XOR – Encryption function
F – Function for calculation of the next key
CPRNG – Chaotic Pseudo-Random Number Generator based on a logistic map

A. *Function F*

Function F was slightly changed compared to the original function from [1] and, for testing purposes, the F function was changed as follows:

$$k_{ni} = \begin{cases} (p_i + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 1 \\ (c_i + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 2 \\ (k_{pi} + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 3 \\ (2 \cdot p_i + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 4 \\ (2 \cdot c_i + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 5 \\ (2 \cdot k_{pi} + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 6 \\ (3 \cdot p_i + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 7 \\ (3 \cdot c_i + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 8 \\ (3 \cdot k_{pi} + d + g + i) \; mod \; 256, \;\; for \; d \; = \; 9 \end{cases} \qquad (1)$$

k_{ni} – Byte of the next key
k_{pi} – Byte of the previous key
c_i – Byte of the last cipher text
p_i – Byte of the previous plain text
g – The last three digits of value x, generated by CPRNG as a natural number on the interval <1.999>
d – The last digit of value x, generated by CPRNG or g
i – The index of the actual byte

III. A FUNDAMENTAL AND INTRODUCTARY COMPARISON

Fundamental modes manipulate the input for the block cipher, and the block cipher is the main "building block" of these modes. Block ciphers also more or less provide the main security elements. This means that fundamental modes are extremely insecure - without an appropriate block cipher. The PM-DC-LM mode tries to become a different mode - and may provide higher "security levels"; as will be shown in the following section.

Fundamental modes modify the input for the next encryption step. The PM-DC-LM mode tries to manipulate the key for the next encryption step, i.e. the Key FeedBack (KFB) mode – which is described in [7]. Compared to KFB mode, the PM-DC-LM mode enables the use of a variable (polymorphous) structure, which is how the key is computed. Both methods are needed to distinguish cipher texts; even if the plain texts are the same.

The main difference between PM-DC-LM and the other fundamental modes is in its polymorphous or "driven" structure. The PM-DC-LM structure is managed by CPRNG. This means that the structure will have been changed with IV and CPRNG adjustments. Different ways are encapsulated in the F function. As a result, one cannot trace a concrete path without knowledge of the input IV or without knowledge of the CPRNG setting. For all the other modes – i.e. the ECD, CBC, CFB, OFB, and CTR modes, it is possible to trace the way these modes run because it is fixed. The way the PM-DC-LM mode runs varies with the last digit of the CPRNG.

It can be stated that the PM-DC-LM mode is not a "proper" mode like the others, since it incorporates higher functionalities -e.g. CPRNG and the F function. The set goal was to design something new - and different.

IV. COMPARISON OF THE IMAGE DATA BASE

Testing was performed using a BMP image file. The image was composed of 320x320 points in Red/Green/Blue (RGB); which represents 320x320x3 bytes of data, plus 54 bytes for the header. Only image data -> 307200 bytes was used for "encryption purposes" and the XOR function was used as the encryption function - even if this is not standard. The image contained 100x100 points of a black colored square on a white background. The length of the key and blocks was 256 bits (32 bytes). The first key and Initialization vector (IV) were different, and were randomly chosen. The following key in hexadecimal format was used:

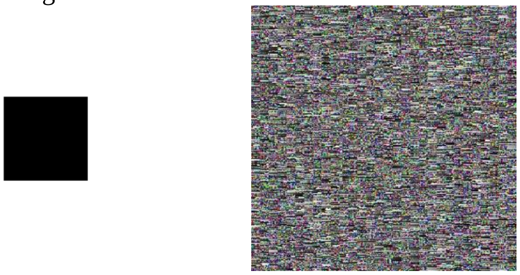
91650ae10ea3ca81d629b0c71dc67d063bf215038025d5750 d2c8c5cf7547787

The following IV in hexadecimal format was used:

fce7cf7ffa651ce5d9d56dd92cc49e13bcc3bd17485d75637a8 00f11aea505c8

A. The PM-DC-LM Mode

The result, after using the designed PM-DC-LM mode, is shown in Figure 2.

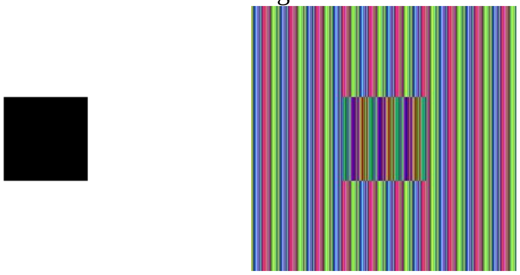


(a) (b)

Figure 2. (a) Original image; (b) Image after using the PM-DC-LM Mode

B. The ECB Mode

Fig. 3 shows the result after using the ECB Mode.

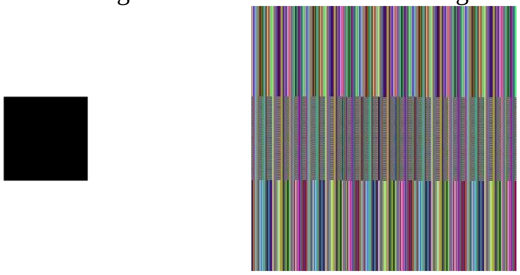


(a) (b)

Figure 3. (a) Original image; (b) Image after using the ECB Mode

C. The CBC mode

The result of using the CBC Mode is shown in Figure 4.

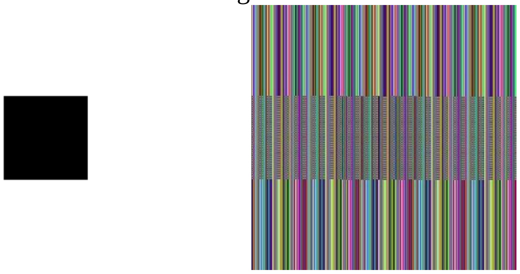


(a) (b)

Figure 4. (a) Original image; (b) Image after using the CBC Mode

D. The CFB Mode shown

Fig. 5 sows the result of using the CFB Mode.

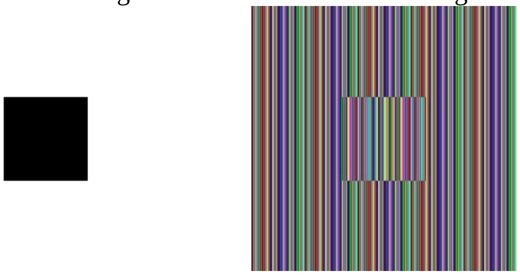


(a) (b)

Figure 5. (a) Original image; (b) Image after using the CFB Mode

E. The OFB Mode

The result of using the OFB Mode is shown in Figure 6.

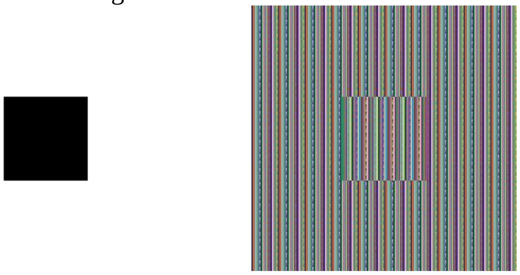


(a) (b)

Figure 6. (a) Original image; (b) Image after using the OFB Mode

F. The CTR Mode

For testing purposes, the CTR Mode was used as follows: the counter was set to number one - represented by a 256-bit number using IV. The result of using the CTR Mode is shown in Figure 7.



(a) (b)

Figure 7. (a) Original image; (b) Image after using the CTR Mode

G. Summary and modification

From the images above, one can see that the CBC, CFB, EBC and OFB modes look more or less alike; while the CTR Mode is a little different; where the difference is in the last bytes of the cipher text blocks. This is because of the counter. Only, the PM-DC-LM Mode - without using a block cipher, looks like “random noise” - excluding blocks with similar shades. These blocks are due to the use of plain text (all zeros or ones in the BMP image) for the calculation of the next key for the encryption phase. As a result, the following key will be calculated using the same bytes. This deficit could be ameliorated by the modification of the first, fourth and seventh equations in the F function by the addition of the byte of the previous cipher text. The F function can be modified as follows:

$$k_{ni} = \begin{cases} (p_i + c_i + d + g + i) \, mod \, 256, & for \; d = 1 \\ (c_i + d + g + i) \, mod \, 256, & for \; d = 2 \\ \left(k_{pi} + d + g + i\right) mod \, 256, & for \; d = 3 \\ (2 \cdot p_i - c_i + d + g + i) \, mod \, 256, & for \; d = 4 \\ (2 \cdot c_i + d + g + i) \, mod \, 256, & for \; d = 5 \\ \left(2 \cdot k_{pi} + d + g + i\right) mod \, 256, & for \; d = 6 \\ (3 \cdot p_i + 2 \cdot c_i + d + g + i) \, mod \, 256, & for \; d = 7 \\ (3 \cdot c_i + d + g + i) \, mod \, 256, & for \; d = 8 \\ \left(3 \cdot k_{pi} + d + g + i\right) mod \, 256, & for \; d = 9 \end{cases} \qquad (2)$$

The result of using the modified PM-DC-LM Mode is shown in Figure 8.

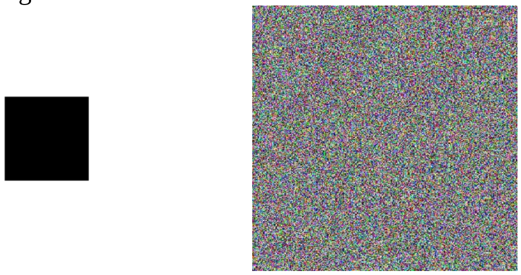


Figure 8. (a) Original image; (b) Image after using the modified PM-DC-LM Mode

This result looks better when compared to the result in Figure 1, since the function was modified by adding cipher text which the F function computes with plain text (for d = 1, d = 4, and d = 7)

V. Speed comparison

This section analyses and compares the time for "encryption" - dependent on the mode. The time depends on the implementation of the mode; but for testing purposes, all modes were implemented in a similar manner. All modes were implemented in Python, Version 3.4 and were written without using parallelizability.

TABLE I. TIME NEEDED FOR "ENCRYPTION" BY MODE

Time for "encryption"			
	BMP 320x320	1 MB Image	10 MB Image
PM-DC-LM	1.163 s	3.786 s	37.858 s
ECB*	0.100 s	0.326 s	3.255 s
CBC	0.182 s	0.592 s	5.924 s
CFB	0.191 s	0.622 s	6.217 s
OFB	0.184 s	0.599 s	5.990 s
CTR*	0.288 s	0.938 s	9.375 s

* Without parallelization

In Table 1, we can see that the PM-DC-LM mode is approximately four times slower than the CTR mode and from six to seven times slower than the CBC, CFB, and OFB modes, and eleven times slower than the ECB mode.

VI. Comparison on the other properties

A. *"Security Level"*

It is a well known fact that the security of an encryption algorithm is mainly based on the block cipher. Thus, the comparison based on this property could be misleading; despite this, one can try to compare it. For testing purposes, the security level will be measured against these conditions:

- Chaining (i.e. the previous block affects the encryption of the next block) - Positive
- IV (i.e. the insertion of another random factor) - Positive
- The level of dependency on block cipher security - Negative
- Extension (i.e. Deterministic Chaos) – Positive

From the conditions above, one could derive that the ECB mode should be the least secure; followed by the CTR mode. The OFB, CBC, and CFB modes follow; all have similar security levels. The PM-DC-LM mode should have the highest security level.

In the PM-DC-LM mode, security depends on the CPRNG and, according to [4], logistic maps may be used as a CPRNG. But this could be changed in an appropriate manner or another type of Deterministic Chaos could be used.

Another important thing is that the IV must be random in order to achieve "indistinguishability" from random bits, used only once.

According to [6], the PM-DC-LM mode has no CCA security; and the mode is secure as a probabilistic encryption scheme. The security level will vary according to the different design of the CPRNG.

No attack using a knowledge of the structure may be used, since the structure cannot be known without a knowledge of the IV or of the CPRNG adjustement. This may be the greatest contribution of this mode.

B. *Errors*

The errors depend upon the place and time of the error occurrence. Only bit errors (where the bit is changed from 0 to 1 or from 1 to 0) during transmission after encryption can be considered. Two sample cases are especially discussed here, (an error in IV, or error in a block of cipher text).

1) *Error bit in IV*

The ECB mode does not have IV. Thus, ECB caused by errors in IV cannot occur. For the CFB and CBC modes, the first block will be decrypted incorrectly whilst in the CTR (where IV is used) and the OFB, and PM-DC-LM modes, all blocks will be decrypted incorrectly.

2) *Error bit in a block of cipher text*

For the ECB, CTR, and OFB modes, the error in a block of cipher text does not affect the decryption of the other blocks. The error will only be in the block corresponding to the block of cipher text with an error bit.

For the CFB and CBC modes, the error in a block cipher affects all of the other decrypted blocks - including any block corresponding to a block of cipher text with an error bit.

For the PM-DC-LM mode, there is a 33.3 % probability that the error bit in the block of the cipher text will not affect the other blocks during decryption.

C. *Parallelizability*

The ECB and CTR modes can be fully parallelized. The CBC and CFB modes can only be parallelized during decryption. The OFB mode cannot be parallelized at all. In PM- DC-LM mode, the CPRNG and encryption can be separately computed and cannot be parallelized.

D. *Affecting the key*

All of the modes compared above - excluding the PM-DC-LM mode, do not affect the key(s) for encryption.

E. *Summary*

All the other properties are summarized in Table 2.

TABLE II. SUMMARIZATION OF THE COMPARISON OF THE OTHER PROPERTIES

Mode	Other properties			
	Parallelizable		Error propagation	Chaining
	Encryption	Decryption		
PM-DC-LM	No	No	Yes	Yes
ECB	Yes	Yes	No	No
CBC	No	Yes	Yes	Yes
CFB	No	Yes	Yes	Yes
OFB	No	No	Yes	Yes
CTR	Yes	Yes	No	No
Mode	Affecting of key	IV/Nonce	The level of dependency on the block cipher security	The level of security without block cipher
PM-DC-LM	Yes	Yes	Low	High
ECB	No	No	High	Very low
CBC	No	Yes	High	Very low
CFB	No	Yes	High	Very low
OFB	No	Yes	High	Very low
CTR	No	Yes	High	Very low

VII. CONCLUSION

In this paper, the authors have tried to compare their own design of a block cipher mode of operation - called PM-DC-LM as an example derived from PM with other modes. Specifically, comparisons were made based on the speed and image data using the XOR function as the encryption algorithm instead of a regular block cipher, even if it is “extra-ordinary”. Using the Advanced Encryption Standard (AES), the result would be different. Additionally, basic properties like parallelizability, “security” level and propagation errors were explored.

The results of these comparisons indicate that the PM-DC-LM mode may be more secure if one uses random IV and an appropriate design for the CPRNG or other PRNG. The authors´ also realized that the PM-DC-LM mode is slower - compared with the other modes, and behaves “randomly”. The PM-DC-LM mode is more prone to error propagation and cannot be parallelized. By changing the F function, the behavior of the mode was changed as was illustrated on the image data.

The PM-DC-LM mode would appear to be a potent mode for an “encryption” algorithm; but the results may be better using block ciphers instead of the XOR function. This mode may be immune to all attacks based on structure.

Since this is a preliminary work, only basic comparisons were made. Future work hopes to continue the research with some interesting findings.

ACKNOWLEDGMENT

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

REFERENCES

[1] P Zacek, R. Jasek, and D. Malanik, “Using the deterministic chaos in variable mode of operation of block ciphers”, in Artificial Intelligence Perspectives and Applications (CSOC 2015), Springer International Publishing, 2015 pp. 347-354, doi:10.1007/978-3-319-18476-0_34.

[2] Current Modes. In: Special Publication 800-38A: First Part: Five Confidentiality Modes 2001. [Online]. Available from: http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf

[3] Modes Development. In: National Institute of Standards and Technology: Computer Security Resource Center [online]. 2001, http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html

[4] J. C. Sprott, Chaos and Time-Series Analysis, Oxford University Press, 2003

[5] R. Senkerik, M. Pluhacek, I. Zelinka, D. Davendra, and Z. Oplatkova, “A brief survey on the chaotic systems as the pseudo random number generators”,, in Interdisciplinary Symposium on Complex Systems, vol 14. Emergence, Complexity and Computation (ISCS 2014),, Springer International Publishing, 2015, pp. 205-214, doi:10.1007/978-3-319-10759-2_22

[6] P. Rogaway, “Evaluation of some block cipher modes of operation”, Feb. 2011. [Online]. Available from: http://web.cs.ucdavis.edu/~rogaway/papers/modes.pdf

[7] J. Hastad and M. Näslund, Key Feedback Mode: a Keystream Generator with Provable Security, Oct. 2000. [Online]. Available from:

http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/kfb/kfb-spec.pdf

[8] H. Lipmaa, P. Rogaway, and D. Wagner. “Comments to NIST concerning AES-modes of operation: CTR-mode encryption”, in Symmetric Key Block Cipher Modes of Operation Workshop, Baltimore, Maryland, US, 2000. [Online]. Available from:

http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/workshop1/papers/lipmaa-ctr.pdf