Establishing Systems Secure from Research with Implementation in Encryption Algorithms

Mikhail Styugin

Research Department, Siberian State Aerospace University
Department of Applied Mathematics and Computer Security, Siberian Federal University
660014 Krasnoyarsk, Russia
(Email: styugin@gmail.com)
(Received June 29, 2016; revised and accepted Nov. 12, 2016 & Jan. 15, 2017)

Abstract

Systems that have a complex technical implementation usually contain many vulnerabilities which cannot be found at the development stage. Security of complex systems can be improved by protecting them from external research. When the operating algorithm of a system remains concealed then it will be more difficult to compromise the system. The present paper reviews a method of modeling information systems, which allows formalizing the amount of information obtained by a researcher. Two methods of establishing systems protected from research are presented. One method is related to complicating the algorithms and the other one is related to their multiplication. Implementation in encryption systems proves fulfillment of cipher security conditions with their modification. Experimental study of the obtained encryptor demonstrated its effectiveness in protecting from many existing types of attacks aimed at block cipher algorithms.

Keywords: Block Ciphers; Cryptography; Indistinguishability; Researcher Model

1 Introduction

Recently methods and technologies for ensuring security of information systems more and more frequently address the objectives to complicate the system research process for an adversary. Evidently, the less information an attacker obtains the less opportunities it has to compromise the system as well as for unauthorized use of the system.

The current trend is also determined by constant increasing complexity of information systems. The complexity of modern information systems does not allow eliminating all potential vulnerabilities and errors at the design stage. The requirement to release a functionally complete application limits time for testing the completed systems. Whereas for an attacker the time for analyzing an application is nominally unlimited. This creates information asymmetry and requires new solutions to be found in information security, solutions to cover the undetected vulnerabilities and errors. Technologies for protecting information systems from external research are the solutions for those problems. One of the prominent trends in that area is, for example, Moving Target Defense [7] technologies. Recently over 150 different MTD techniques [14] related to LAN security [2], protection from program code injection [9], protection from XSS attacks [15], protection from DDoS attacks [10], etc.

The present paper considers protection of a system from research at the level of algorithms which is implemented in encryption algorithms. This area was chosen because currently requirements to encryption algorithms are the most formalized ones. The problem we have studied is about whether it is possible to formalize a system researcher and to conclude that system research security problem can be solved by using the model obtained.

Security of ciphers themselves is not a new problem. It seems obvious that with an unobservable encryption algorithm it will be more difficult for an adversary to accomplish a ciphertext attack [5, 11]. For instance, papers [6, 12, 19] consider modifications of symmetric encryption algorithms performed by changing rules of permutation and substitution. Paper [20] considers modification of the mode of operation so that the method for presentation of the next cipher block depends on the parameters obtained at the previous step. Cipher modification is also effective in security against side-channel attacks [17] which remain effective also for existing symmetry encryption standards such as AES.

The drawback of all the studied cipher modification solutions is that the cipher variation method is strictly defined. The knowledge of the method simplifies system analysis. A completely research secure system shall not disclose any information related to the methods of cipher text generation.

2 Formalization of a System Researcher

The researcher model is assumed as in paper [18]. The research target remaining to be a black box, the sufficient modifications shall be made as follows.

Take a tuple of three values (x, y, z). Value x stands for the number (share) of observable input values, value y is the observable number of functions of the black box and z is the number of observable output values. A completely observable box is $(1, 1, 1)$, a completely unobservable box is $(0, 0, 0)$. When the black box's output can be observed then it is denoted as $(0, 0, 1)$. Whereas the set of different input values and function values consists of N elements and the system researcher has a function for one of them with which it transforms into the required output value then the box is denoted as $(1/N, 1/N, 1)$.

An encryption system which does not disclose any information can be conceived as two boxes:

$$(0, 0, 1) \rightarrow (1, 0, 0).$$

The first block is denoted as A, hence the second block being reverse to the first one is denoted as A^{-1}. The notation for the obtained system is AA^{-1}.

In order to find the possibilities for researching the system the second level of variables can be introduced, which indicates that the researcher has the data presentation method. For example, one-time pads will have the following notation:

$$(0_1, 0_1, 1) \rightarrow (1, 0_1, 0_1).$$

That implies that even without having the final transformation function we know that it is denoted as $c = m \oplus k$. Hence, having the only one value and one function for one-time pad will enable to disclose all information about the system

$$((1/N)_1, (1/N)_1, 1) = (1, 1, 1).$$

For box $(0, 0, 1)$ in which the function's construction principle is unobservable (for example when the transformation implements a purely random function), then the attempt to obtain information on the only input value and the function will not enable to obtain any additional data:

$$(1/N, 1/N, 1) \neq (u, v, 1), \text{ where } u, v > 1/N.$$

Similarly when the system's function is an instance of a more general functionality then the following tuple can be defined:

$$(0_{0_1}, 0_{0_1}, 1).$$

By the analogy to the encryption in that notation we obtained the algorithm for algorithm generation.

3 Algorithm Research Security by Blurring

When the state of the researched system $(..., 1, ...)$ is unacceptable then the system can be transferred into state $(..., 0_1, ...)$. In case the latter state is also unacceptable then the system can be transferred into state $(..., 0_{0_1}, ...)$. The above procedure shall be called "blurring" of the box's properties.

Absolute ciphers in cryptography are not always a practical structure therefore information obtained by a researcher can be expressed as negligibly small values $\epsilon(n)$, which depend on some parameters as key length n, for example.

A symmetric cipher can be denoted as the following scheme:

$$(\epsilon(n)_1, \epsilon(n)_1, 1) \rightarrow (1, \epsilon(n)_1, \epsilon(n)_1).$$

Algorithm of the above cipher can be "blurred" by defining the algorithm for selecting the encryption algorithm. Then the following scheme is obtained:

$$(..., \epsilon(n)_{\epsilon(n)_1}, 1) \rightarrow (1, \epsilon(n)_{\epsilon(n)_1}, ...).$$

Let us consider an example. Given an AES algorithm, Algorithm 1 shall be used instead of the typical substitution table.

Algorithm 1 Substitution function

1: **Function** SubBytes (t: $0...2^8 - 1$, k:integer)
2: $a = t - 1 \bmod 28$;
3: $b_i = a_{(k+i) \bmod 8} \oplus a_{(k+i+4) \bmod 8} \oplus a_{(k+i+5) \bmod 8} \oplus a_{(k+i+6) \bmod 8} \oplus a_{(k+i+7) \bmod 8} \oplus (k_{(k+i) \bmod 8} \bmod 2^8)$;
4: $result = b$;

The above function substitutes an input value t for value b. It performs the substitution based on key k. Depending on the key value the *SubBytes* function generates 28! (factorial) substitution tables. That number is so high that we can effectively use keys of 128 bits or 512 bits as input with a negligibly small probability that substitution tables may repeat. Hence, now the system has two keys. One of the keys is used for generating substitution tables and the other one is a regular AES key. When the length of the key for selecting substitution tables equals m then upon analyzing the ciphertexts an adversary is not able to distinguish a substitution table with an accuracy greater than a negligibly small value $\epsilon(m)$.

Similarly the cipher can be blurred further by introducing the function for a function's modification:

$$(..., \epsilon(n)_{\epsilon(m)_{\epsilon(h)_1}}, 1) \rightarrow (1, \epsilon(n)_{\epsilon(m)_{\epsilon(h)_1}}, ...).$$

According to the Kerckhoffs' principle [8] a system's operation algorithm shall be open. In our case the principle is maintained, but the adversary's knowledge of the system is always moved to the last level of the scheme (Figure 1).

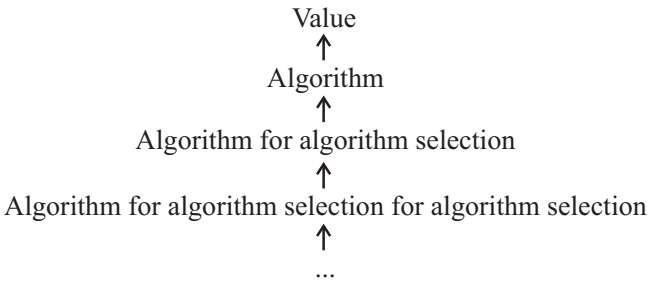

Figure 1: "Blurring" of the box's properties

Following the Kerckhoffs' principle requires that each tuple shall end with 1. Violation of the Kerckhoffs' principle, i.e. when the tuple ends with 0. Then the parameter is defined as absolutely unobservable.

Theorem 1. *When random variables k_1, k_2, ..., k_m with a bit length of $\|k_1\| = n_1$, $\|k_2\| = n_2$, ..., $\|k_m\| = n_m$ are used for creating an encryption algorithm and the cipher scheme is $(..., \epsilon(n_1)_{...\epsilon(n_m)_1}, 1) \rightarrow (1, \epsilon(n_1)_{...\epsilon(n_m)_1}, ...)$, then the probability for distinguishing the input state is less or equals to negligible value $\epsilon(n_1) \cdot \epsilon(n_2) \cdot ... \cdot \epsilon(n_m)$.*

Proof. For proving the above statement we should consider the fact that to find the required algorithm at step q, $\epsilon^{-1}(n_{q+1})$ operations at step $q+1$ must be performed for the algorithm for selecting an algorithm. As operations are performed consecutively, hence the total number of operations would be $\epsilon^{-1}(n_q)\epsilon^{-1}(n_{q+1})$. Thus, using the method of mathematical induction it is proved that the statement, which declares that the probability of distinguishing input in one operation would be less or equal to $\epsilon(n_1) \cdot \epsilon(n_2) \cdot ... \cdot \epsilon(n_m)$, is true. □

4 Research Security by Multiplication of Algorithms

Besides blurring a specific algorithm, multiplication of algorithms can also be implemented. The multiplication would be a sequence of direct and reverse boxes. The previous sections of the paper only simple schemes were considered, which are denoted as A for a separate algorithm and AA^{-1} for encryption systems.

Algorithm multiplication would be a consecutive recording of $ABCD...$, where every following algorithm has the output of the previous algorithm as its input as well as some set of properties. It is assumed that all algorithms are executed in polynomial time.

When one of the algorithms in the sequence has some indistinguishability property, then the whole sequence has the indistinguishability property provided that the parameters must not be reused. The statement can be formalized and proved by separate examples. Below are the examples as applied to cipher area.

Theorem 2. *When encryption scheme AA^{-1} has the indistinguishability parameter with respect to input data, then the parameter is included in any scheme where notation is in the form of $A_1...A_uAA_{u+1}...A_mA_m^{-1}...A_{u+1}^{-1}A^{-1}A_u^{-1}...A_1^{-1}$ provided that the parameters used in A are not used in the other algorithms of the scheme.*

Proof. The indistinguishability definition for encryption algorithms is applied as presented in [13]. An attacker provides a pair of messages m_0 and m_1 of equal length. The encryption algorithm gets a random number of message $b \leftarrow \{0,1\}$ and the message with the number is encrypted $c \leftarrow Enc_k(m_b)$. The obtained ciphertext is sent to the attacker (I) in it shall find the number of the encrypted message b' and in case $b' = b$ then the experiment is considered to be accomplished $PrivK_{I,\Pi}^{eav}(n) = 1$, otherwise $PrivK_{I,\Pi}^{eav}(n) = 0$. The encryption scheme is indistinguishable when there is such a negligibly small function $negl$ for all probabilistic polynomial time attackers I so that the following condition is fulfilled:

$$Pr[PrivK_{I,\Pi}^{eav}(n) = 1] \leq \frac{1}{2} + negl(n).$$

It is evident that operation of algorithms $A_1...A_u$ will have no impact on the indistinguishability condition as their output is input for algorithm A, for which the indistinguishability condition is fulfilled.

For algorithms $A_{u+1}...A_m$ it shall be noted that according to the theorem's conditions they have parameters that are different from the parameters of A. A proof by reduction shall be performed. Assume there are algorithms $A_{u+1}...A_m$ that are such that in conjunction with algorithm A, i.e. $withAA_{u+1}...A_m$ a distinguishable scheme will be obtained for which the following is fulfilled

$$Pr[PrivK_{I,\Pi}^{eav}(n) = 1] > \frac{1}{2} + negl(n).$$

As a result a probabilistic-polynomial time algorithm $A_{u+1}...A_m$ was obtained. The algorithm can distinguish output of algorithm A, which contradicts the indistinguishability condition of algorithm A. □

Similar conditions can be established for the other cipher indistinguishability requirements as well.

Theorem 3. *When encryption scheme AA^{-1} is CPA-secure, then every other scheme that has notation in the form of $A_1...A_uAA_{u+1}...A_mA_m^{-1}...A_{u+1}^{-1}A^{-1}A_u^{-1}...A_1^{-1}$ also has that property on condition that parameters used by A shall not be used by the other algorithms of the scheme.*

Proof. The proof is similar to Theorem 2. □

Theorem 4. *When encryption scheme AA^{-1} is CCA-secure then every other scheme that has notation in the form of $A_1...A_uAA_{u+1}...A_mA_m^{-1}...A_{u+1}^{-1}A^{-1}A_u^{-1}...A_1^{-1}$ also has that property on condition that parameters used by A shall not be used by the other algorithms of the scheme.*

Proof. The proof is similar to Theorem 2. □

Multiplication of cracking difficulty of ciphers in consequent implementation of their algorithms can be considered.

Theorem 5. *When encryption scheme AA^{-1} fulfils the indistinguishability condition which is expressed in the requirement $Pr[PrivK^{eav}_{I,A}(n) = 1] \leq \frac{1}{2} + negl(n)$ and encryption scheme BB^{-1} fulfils the indistinguishability condition which is expressed in the requirement $Pr[PrivK^{eav}_{I,B}(m) = 1] \leq \frac{1}{2} + negl(m)$ then encryption scheme $...A...B...B^{-1}...A^{-1}...$ shall fulfill the indistinguishability requirement expressed as follows $Pr[PrivK^{eav}_{I,...A...B...B^{-1}...A^{-1}...}(n,m) = 1] \leq \frac{1}{2} + negl(n)negl(m)$*

Proof. Proof similar to Theorem 1. □

The section below presents a practical implementation of algorithms with blurring and multiplication.

5 Establishing Research Secure Systems

Let us review at a practical example. Assume that a symmetric encryption scheme should be established for message exchange between two users. It is possible to generate a random sequence of 128 to 2048 bit for one session and provide the random sequence to both users. The classic symmetric encryption scheme requires a strictly set key size and a predefined algorithm. However, we want to establish an encryption system to generate a different algorithm every time by efficiently using the whole random sequence. Then the practical impossibility for an adversary to distinguish either the encrypted message or the algorithm implemented, does not allow performing any computational attacks or side-channel attacks.

Assume that the encoder program can be a $C \in \mathbf{C}$ algorithm. Then the cardinality of set $\mathbf{C}$ has to be large enough to use the provided random sequence. When it is possible to generate a 4096 bit sequence then the number of elements in set $\mathbf{C}$ shall be greater than 2^{4096}. It was demonstrated above that it is easy to accomplish by only correcting the encoder's substitution tables.

A basic requirement to a research secure algorithm. *An adversary shall not be able to compute the implemented $C \in \boldsymbol{C}$ algorithm with accuracy greater than the negligibly small function from the length of the random sequence, which is $negl(u)$.*

In order to create an encryptor that is guaranteed to fulfill the indistinguishability requirements, the cipher's research protection algorithm shall be divided in two operations. The first operation implies consequent multiplication of separate algorithms (boxes) with the predefined features indistinguishable (IND), CPA-secure and CCA-secure. At this stage a transformation sequence with the defined requirements and guaranteed multiplication of difficulty is established. Then each box is blurred with the operations of permutation and substitution, while the difficulty multiplication requirements are no longer applied to them. The encryptor scheme is shown in Figure 2.

The obtained encryption algorithm complies with the requirements of IND-CCA, IND-CPA and IND for adversaries that are able to perform up to $2^{\|k\|}$ operations.

Algorithm 2 is the encryptor mechanism at the pseudocode level. In Theorems 2, 3 and 4 above it was established that for fulfilling the requirements applied to the encryption algorithm the key cannot be reused in every "box". Therefore, function cut is introduced which cuts the random sequence in pieces of specified length as required.

Algorithm 2 Cipher

```
Function cipher (k, m)
Cut ← n bit
i = 0
while ∃k_cur = cut(i, k) do
    m = A_{hash(k_cur) mod m}(k_cur, m)
    i = i + 1
end while
result = m
```

In the above case, algorithm A_j is launched pseudorandomly (by function hash). Each of the algorithms $A_0, ..., A_{m-1}$ is a typical box with the confirmed requirements of IND, CPA and CCA. Thus, function cipher performs multiplication of algorithms as a sequence of boxes $A_1...A_uA_u^{-1}...A_1^{-1}$. Then algorithm blurring is performed in each box to establish research security as shown below

$$(..., \epsilon(n_1)_{...\epsilon(n_m)_1}, 1) \rightarrow (1, \epsilon(n_1)_{...\epsilon(n_m)_1}, ...).$$

The above blurring scheme is demonstrated in the example of the pseudocode in Algorithm 3.

Algorithm 3 An example of the pseudocode

```
Function A_j (k, m)
while hash(k) ≠ const do
    Gen func_sub, k
    Gen func_per, k
    m = Item(func_sub, func_per, k)
    k = func_sub(k)
    k = func_per(k)
end while
result = m
```

The above function runs until $hash(k)$ equals the predefined constant. The constant is defined by experiment, when the number of the algorithm blurring steps can be considered sufficient. The key is used for generating the unique substitution function $func\_sub(k)$ and the unique permutation function $func\_per(k)$. Then a block cipher with the obtained functions is launched. After that the function of substitutions and permutations is applied to the key itself and the cycle is repeated.

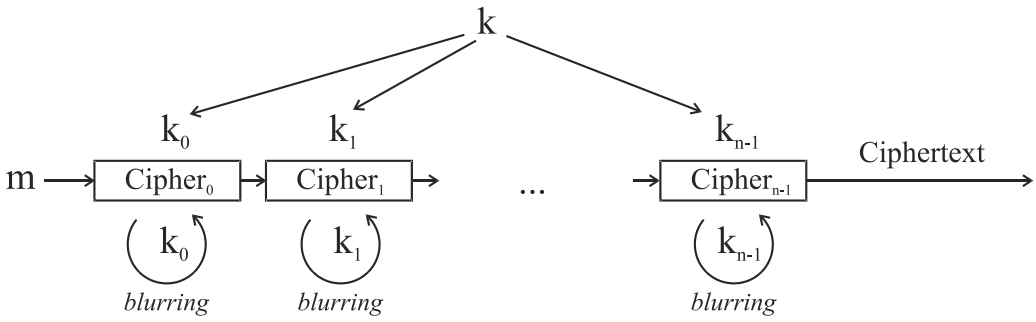

Figure 2: The research secure encryptor scheme

Practical implementation of the research security program was calculated hash functions only once and then it generated the code compliant to the algorithm's functionality to speed up the cipher execution.

6 Experimental Research

In order to perform experimental research of the research secure algorithm, the program mentioned in the previous paragraph has been developed to implement the multiplication algorithm and blurring algorithm. The researched algorithm was a block cipher. Advanced Encryption Standard (AES), International Data Encryption Algorithm (IDEA) and the GOST Russian encryption standard were used as boxes. It should be noted that while substitution and permutation tables are strictly defined in AES and IDEA, in GOST the algorithm blurring procedure can be carried out with these parameters without exceeding the limits of the standard.

All above algorithms comply with the indistinguishability requirements (IND), indistinguishability under chosen-plaintext attack (CPA-secure) and indistinguishability under chosen ciphertext attack (CCA-secure). In order to simplify the block assembly, key length of 128 bit was used as all the provided ciphers can operate with that key length.

A similar function was used as generator of substitution and permutation tables in Algorithm 1 with an input key as the parameter. The range of permutations and substitutions is defined by the type of the block cipher used. The generated cipher was tested for susceptibility to algebraic attacks that decrease the number of enumerating operations [1], algebraic attacks of side channels [13, 16] and differential attacks to decrease cipher cracking difficulty by many orders involving a large volume of memory [3, 4].

The obtained algorithm on a 128 bit random sequence demonstrated failure of all the above attack classes. However, multiplication of cracking difficulty with a longer sequence cannot be tested in practice.

7 Conclusions

The present paper provided a general approach to research security of program algorithms based on two methods. The first method is algorithm blurring by constant shifting the researcher's visibility point. Instead of getting the algorithm, a researcher can only get an algorithm for algorithm generation or an algorithm for generation of an algorithm for generation of an algorithm, etc. That shift enables making the system more complex by introducing additional parameters of randomness or pseudorandomness. The second method is based on algorithm multiplication. Its special feature is that it can be used for expanding any of the indistinguishability properties of its individual components to the whole algorithm.

Theoretic and experimental study of the methods as applied to encryption algorithms demonstrated their effectiveness against many existing attacks aimed at block ciphers and involve algebraic analysis and exploitation of side channels.

Acknowledgments

This study was funded by RFBR according to the research project No. 16-29-09456 ofi_m and Grant of RF President (MK-5025.2016.9).

References

[1] A. Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique cryptanalysis of the full AES," in *Proceedings of The 17th International Conference on the Theory and Application of Cryptology and Information Security*, pp. 344–371, Seoul, Korea, Dec. 2011.

[2] M. Carvalho and R. Ford, "Moving-target defenses for computer networks," *IEEE Security and Privacy*, vol. 12, no. 2, pp. 73–76, 2014.

[3] N. T. Courtois, "An improved differential attack on full gost," Tech. Rep. IACR, 2012.

[4] N. T. Courtois and M. Misztal, "Differential cryptanalysis of gost," Tech. Rep. IACR, 2011.

[5] T. Gulom, "The encryption algorithm GOST28147-89-PES16-2 and GOST28147-89-RFWKPES16-2,"

International Journal of Electronics and Information Engineering, vol. 6, no. 1, pp. 1–11, 2017.

[6] M. I. Husain, K. Courtright, and R. Sridhar, "Lightweight reconfigurable encryption architecture for moving target defense," in *Proceedings of The IEEE Military Communications Conference (MILCOM'13)*, pp. 214–219, San Diego, USA, Nov. 2013.

[7] S. Jajodia, A. K. Ghosh, V. Swarup, C. Wang, and X. S. Wang, *Moving Target Defense. Creating Asymmetric Uncertainty for Cyber Threats*, Advances in Information Security, Springer, 2011.

[8] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, Boca Raton: Chapman and Hall/CRC, 2007.

[9] P. Larsen, S. Brunthaler, and M. Franz, "Automatic software diversity," *IEEE Security and Privacy*, vol. 13, no. 2, pp. 30–37, 2015.

[10] D. Ma, Z. Xu, and D. Lin, "Defending blind ddos attack on sdn based on moving target defense," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering*, vol. 152, no. 1, pp. 463–480, 2015.

[11] A. Mersaid, T. Gulom, "The encryption algorithm AES-RFWKIDEA32-1 based on network RFWKIDEA32-1," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 1–11, 2016.

[12] H. Mestiri, F. Kahri, B. Bouallegue, and M. Machhout, "A high-speed aes design resistant to fault injection attacks," *Microprocessors and Microsystems*, vol. 41, no. 1, pp. 47–55, 2016.

[13] Mohamed Saied Emam Mohamed, Stanislav Bulygin, Michael Zohner, Annelie Heuser, and Michael Walter, "Improved algebraic side-channel attack on AES," *Journal of Cryptographic Engineering*, vol. 3, no. 3, pp. 139–156, 2014.

[14] H. Okhravi, T. Hobson, D. Bigelow, and W. Streilein, "Finding focus in the blur of moving-target techniques," *IEEE Security and Privacy*, vol. 12, no. 2, pp. 16–26, 2014.

[15] J. Portner, J. Kerr, and B. Chu, "Moving target defense against cross-site scripting attacks," *Lecture Notes in Computer Science*, vol. 8930, pp. 85–91, 2015.

[16] M. Renauld, F. X. Standaert, and N. V. Charvillon, "Algebraic side-channel attacks on the AES: Why time also matters in DPA," *Lecture Notes in Computer Science*, vol. 5747, pp. 97–111, 2009.

[17] W. Shan, L. Shi, X. Fu, X. Zhang, C. Tian, Z. Xu, J. Yang, and J. J. Li, "A side-channel analysis resistant reconfigurable cryptographic coprocessor supporting multiple block cipher algorithms," in *Proceedings of The 51st Annual Design Automation Conference*, pp. 1–6, San Francisco, United States, June 2014.

[18] M. Styugin, "Protection against system research," *Cybernetics and Systems: An International Journal*, vol. 45, no. 4, pp. 362–372, 2014.

[19] Y. Wang, L. Wang, R. Yao, Z. Zhang, and C. Jiang, "Dynamically reconfigurable encryption system of the AES," *Cryptography. Wuhan University Journal of Natural Sciences*, vol. 11, no. 6, pp. 1569–1572, 2006.

[20] P. Zacek, R. Jasek, and D. Malanik, "Using the deterministic chaos in variable mode of operation of block ciphers," *Artificial Intelligence Perspectives and Applications. Series Advances in Intelligent Systems and Computing*, vol. 347, no. 1, pp. 347–354, 2015.

Biography

Mikhail Styugin is a senior lecturer at Siberian Federal University and a scientist at Siberian State Aerospace University (Krasnoyarsk, Russia). He holds a PhD degree in computer science. He conducts research in area of information security system and technologies of information warfare. He owns two companies that develop solutions in area of information security system in the Internet .