Hardware Tools for Pseudonondeterministic Block Ciphering

Yurii Baryshev
dept. of Information Protection
Vinnytsia National Technical University
Vinnytsia, Ukraine
yuriy.baryshev@gmail.com

Апаратні Засоби для Псевдонедетермінованого Блокового Шифрування

Юрій Баришев
кафедра захисту інформації
Вінницький національний технічний університет
Вінниця, Україна
yuriy.baryshev@gmail.com

***Abstract*—Analysis of known approaches of block ciphering designing is presented at this paper. Implementation of the pseudonondeterministic cryptography was proposed to avoid determined drawbacks of known block cipher designing approaches. The automata model was shown to formalize the pseudonondeterministic block ciphering process. The method based on the model was developed. Both method and model allowed to design the structure of specialized processor for pseudonondeterministic ciphering. The structure's blocks were adapted for FPGA implementation.**

***Анотація*—В даній роботі представлено аналіз відомих підходів до проектування блочного шифрування. Запропоновано реалізацію псевдонедетермінованої криптографії для усування виявлених недоліків відомих підходів до проектування блочних шифрів. Наведено модель автомата, яка формалізує процес псевдонедетермінованого блокового шифрування. Розроблено метод, заснований на моделі. Як метод, так і модель дозволили спроектувати структуру спеціалізованого процесора для псевдонедетермінованого шифрування. Представлено блоки, що входять до даної структури, адаптовані для реалізації за допомогою ПЛІС.**

Keywords—cipher; encryption; pseudonondeterministic; automaton; specialised processor.

Ключові слова—шифр; зашифрування; псевдонедетермінований; автомат; спеціалізований процесор.

I. INTRODUCTION

There are known two main approaches of cipher designing according to the algorithm openness to the public: proprietary and open ones. History shows that latter ones are more attractive for the implementations. The obviousness of this was noticed even at XIX century [1], when basic principal of modern cryptography was postulated: the cryptography transformation infeasibility should be gained by keeping in secret key information, but not algorithms, which implements the transformation. Moreover recent cryptography transformation development methodology discourages researches of the proprietary approach. For instance, modern proprietary algorithms are to be considered inadequate for most application because of both the reasons described in [1] and lack of scientific resources amount needed for such algorithms proper development and analyses [2, 3].

Despite these concerns, there are a lot of drawbacks caused by the cryptographic transformation algorithms openness too. And ciphers are among the most vulnerable ones. For instance, the DES algorithm openness makes it to become object of both differential cryptanalysis [4] and reduced rounds ciphers analysis [5]. It should be pointed out, that the latter attack is direct consequence from the algorithm openness. Thus, open ciphers designing approach despite many positive peculiarities possesses drawbacks. Therefore, known approaches development shouldn't be considered as complete.

There is known pseudonondeterministic approach of cryptographic transformations development, which can aid in further development of ciphers designing methodology [6-8]. In particular, approach of pseudonondeterministic hashing implementation was presented in [7, 8], which allow to gain infeasibility increasing against generic attacks. So the similar effect is to be gained through the implementation of the approach for the block ciphers.

However a lot of tasks are to be solved to implement the approach. The similarity between block ciphering and hashing

from the point of view presented in work [6] encourages further research at this direction. That's why performing pseudononderministic block ciphers development is important.

The goal of this research is to improve block ciphers infeasibility against cryptanalysis based on knowledge of ciphering algorithm using hardware tools development for the pseudononderministic approach implementation.

The following tasks are to be solved to reach the goal:

- formalization of a pseudononderministic ciphering approach;
- ciphering method development according to the approach;
- pseudononderministic automaton structure development.

II. Pseudononderministic Block Ciphering

Consider automata theory to formalize pseudononderministic approach. It is known, that the automaton is a subject, which receives symbols of the subset $A^* \subseteq A$, where A is a finite alphabet [3, 9, 10]. For such finite alphabet an automaton is formalized as the following set of five [9]:

$$Automaton\left(A, S, s_0, T, F\right), \tag{1}$$

where S – a set of automaton states; s_0 – automaton's initial state $\left(s_0 \in S\right)$; T – a set of allowed final states; F – a transition rule.

It is known that automatons are divided into deterministic and nondeterministic ones depending on the transition rule [3, 9, 10]. The next state of deterministic one is easy-to-anticipate; while the transition rule of the nondeterministic is obscured. The principal idea behind the pseudononderministic cryptography is that the transformations should look like ones performed by nondeterministic automaton for the intruder, while they remain being deterministic for a person, who knows a secret (a key) [6, 8]. Such conception allowed to improve hash functions infeasibility [7, 8], consequently the similar result is expected for the block ciphering. The automaton for pseudononderministic ciphering was proposed in work [6]:

$$PNDC = \left\{\left\{m_i\right\}, \left\{\varepsilon_{k_i v_i}\left(m_i\right)\right\}, k, \varepsilon_{k_i v_i}\left(m_l\right), E, V\right\}, \tag{2}$$

where $\left\{m_i\right\}$ – a set of possible plain texts; $\left\{\varepsilon_{k_i v_i}\left(m_i\right)\right\}$ – set of possible ciphertexts; k – a key; E – a set of possible encryption algorithms $\varepsilon_{k_i v_i}(\cdot)$; V – a set of control vectors v_i.

The main difficulty of model (2) implementation is transformation development. The following one is proposed to be used for the cause:

$$x \underset{a,b,c}{\oplus} y = ax + by + c\left(mod\ n\right), \tag{3}$$

where $\left(a, b, c\right)$ is a part of control vector v_i and $a \neq 0$, $b \neq 0$, $a < n$, $b < n$, $c < n$.

The method of pseudononderministic block ciphering *PNDC* (2) based on the operation (3) could be performed by the following steps:

- the message M is split into the l blocks of q bit length. In the case, when the message length cannot be divided without a remainder by q it is to be padded by random number of this remainder's length;
- the key is divided into two parts for initialising key and control vector scheduling procedures respectively;
- ith message block m_i ($i = \overline{1, l}$) is processed iteratively;
- after the last lth message block m_l is processed the ciphering process stops.

Eeach ith message block m_i ($i = \overline{1, l}$) should be processed iteratively in the following way

- the ith control vector is computed v_i;
- the ith subkey is yielded k_i;
- ciphering transformation $\varepsilon_{k_i v_i}(\cdot)$ is chosen;
- the ciphertext e_i is yielded:

$$e_i = \varepsilon_{k_i v_i}\left(m_i\right); \tag{4}$$

- after finishing of ith message block m_i processing the yielded ciphertext e_i is chained with the next (i+1)th iteration accordingly with the block cipher mode of operation, which is determined by terms of a particular task, that is solved by ciphering.

The ciphering transformation $\varepsilon_{k_i v_i}(\cdot)$ at (4) is proposed to perform by at least $t = q / log_2 n$ rounds based on the operation (3). For instance, consider the following round transformation:

- the ith subkey k_i and the ith message block m_i are split into t parts $k_i = k_{i1} || k_{i2} || \ldots || k_{it}$, $m_i = m_{i1} || m_{i2} || \ldots || m_{it}$;
- according to control vector v_i value the parameters a_i, b_i and c_i are chosen;
- the following operation is performed:

$$m_{ij} = m_{ij} \underset{a_i, b_i, c_i}{\oplus} k_{ij}; \tag{5}$$

- the permutation of bits within m_i is performed (for instance, cyclic shift leftwards for $log_2 n$ positions) to

achieve impact of each subkey bit for resulting ciphertext value.

The value of m_i yielded after the last round is the *i*th part of the ciphertext e_i.

III. Specialised Processor Structure

The most efficient implementation from the performance point of view of any computational method is hardware one. The automata model (2) provide the ability to obtain the hardware implementation using synthesis method [3, 10]. Therefore the structure of specialized processor, which implements pseudonondeterministic ciphering is proposed to apply ideas described above (fig.1).

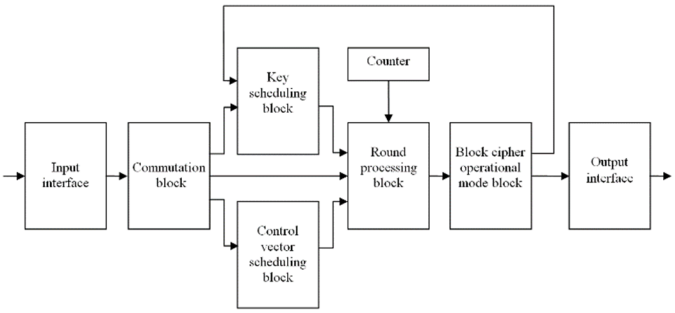

Fig. 1. Structure of pseudonondeterministic ciphering processor

The processor presented on the fig. 1 performs its task in the following way:

- initializing ciphering parameters by transferring key data to the key and control vectors scheduling blocks via input interface and commutation block;
- each time a data block is processed the counter is set accordingly with the round number and the number is decreased after completion of each round computation;
- each time the message block is transferred to the input interface it is further transferred to the round processing block, where the transformation (3) and permutation are performed;
- yielded result of round transformation would remain at the round processing block until the counter value differs from zero;
- after ciphering iteration is completed the ciphertext e_i is chained according with the user defined block cipher operational mode, which might draw to change of (i+1)th subkey value;
- yielded ciphertext e_i is transferred to the user by output interface.

The principal element of the processor structure (fig. 1) is the round processing block. The instance of this block internal structure is presented on fig. 2.

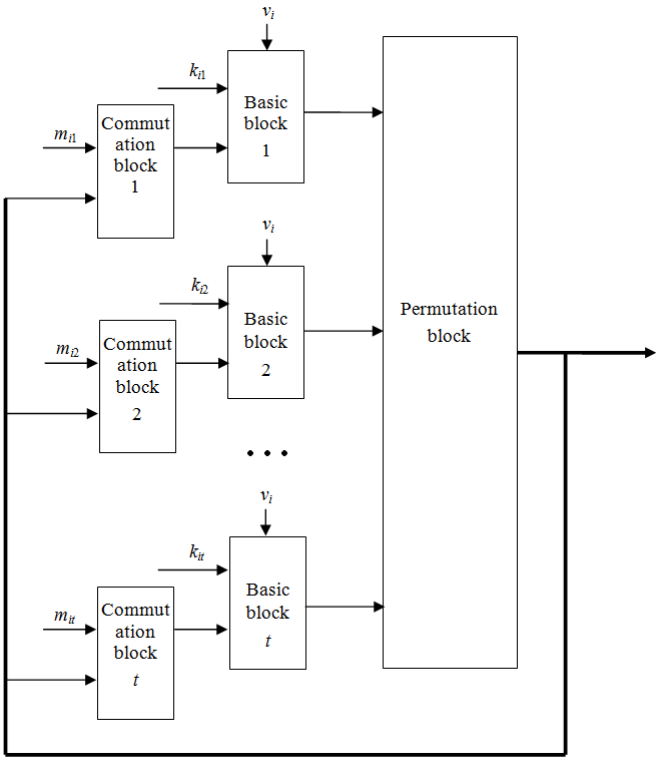

Fig. 2. Structure of the round processing block

Commutation blocks of the structure shown on fig. 2 provide *i*th message block m_i inputting to respective basic block at the implementation of the first round performance at the iteration. A basic block is one, which performs operation $\underset{a,b,c}{\oplus}$ for any allowed set of three (a,b,c) values. Outputs of basic blocks are sent to the permutation block, which can be implemented by a cyclic shift register. The permutation block output is divided into t parts, which are used as basic blocks input at the next round. After all rounds were performed the *i*th part of the ciphertext e_i is yielded.

The basic block structure can be implemented is several ways. One of them is received by the operation $\underset{a,b,c}{\oplus}$ performing. The implementation is presented on fig. 3.

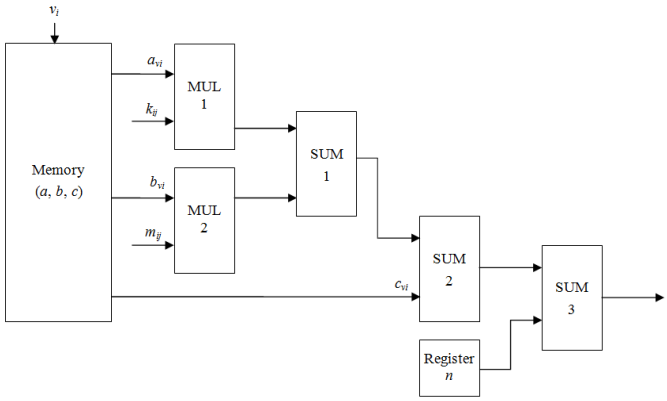

Fig. 3. The basic block implementation based on arithmetic computations

It is obvious, that hardware implementation of the transformation (3) is presented on fig. 3. The block performs in the following way:

- on the basis of control vector v_i value the three (a, b, c) , which are to be used at the iteration, is determined;
- the series of the multiplications and additions are performed by blocks MUL 1, MUL 2, SUM 1 and SUM 2;
- determination of the remainder modulo n is performed by the block SUM 3.

Another implementation of the basic block is based on Cayley's tables. In this case principal element of the structure is memory organized as according table operation device's architecture, when operands are interpreted as memory cells addresses. Such elements within basic block architecture presented on fig. 3 are called "Block (a_z, b_z, c_z)", where $z = \overline{1, \eta}$ (η – the number of threes (a, b, c)).

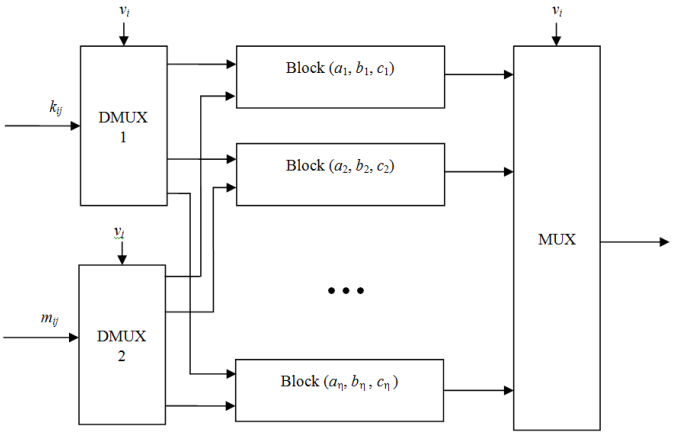

Fig. 4. The basic block implementation based on Cayle's tables

Either of basic block implementations could be used for the pseudonondeterministic ciphering processor. The decision, which of them should be used for the particular task solving, is to be drawn bearing in mind the memory cell implementation of the particular solution. Therefore if there is no restriction for the used memory and its output yielding rapid enough the on presented on fig. 4 should be used. In the case of hardware complexity minimization necessity the one shown on the fig. 3 seems to be more preferable for implementing. Still the threshold is heavily depends on used production technologies.

The structure of these blocks were developed considering both pure hardware implementation and one using VHDL language for the programming FPGA. Presented blocks consist of the basic digital components of computer systems, which aids pure hardware implementation. At the time structure of blocks is based on repetitive using of the similar objects, those could be described by the VHDL means. At the research the latter implementation was chosen to perform, because it provides more flexibility into future development of the presented designs and allows to perform experimenting for its features studying and optimization using alterations of its structure elements.

CONCLUSIONS

Performed analysis of the modern cryptography transformation designing showed the necessity of their further development, because all of them contain drawbacks, which are used by intruders for cracking the transformations. It was shown how this results in block ciphers development.

The usage of pseudonondeterministic approach was proposed for block ciphers infeasibility improving. The automata model based on such approach was presented, which allowed its hardware implementation using synthesis method. The proposed structure of specialized processor provides ability of its optimization for the performance or used memory parameters. This processor structure was designed considering its implementation both using FPGA and pure hardware.

ЛІТЕРАТУРА REFERENCES

[1] A. Kerckhoffs. "Military cryptography", *Journal of Military Sciences,* 9(1): 5–38, 1883. (in French) Available: http://www.petitcolas.net/kerckhoffs/crypto_militaire_1.pdf

A. Kerckhoffs. "La cryptographie militaire" *Journal des Sciences Militaires*, 9(1):5–38, 1883. Available: http://www.petitcolas.net/kerckhoffs/crypto_militaire_1.pdf

[2] R. Verdult. "The (in)security of proprietary cryptography." PhD thesis. KU Leuven, Radboud Universiteit Nijmegen, 2015, 274 p.

[3] V. V. Skobelev, V. G. Skobelev "Ciphersystems Analyses" Donetsk, IAMM NAS of Ukraine, 2009, 479 p. (in Russian)

В. В. Скобелев, В. Г. Скобелев "Анализ шифросистем", Донецк, ИПММ НАН Украины, 2009, 479 с.

[4] Biham E., Shamir A. "Differential Cryptanalysis of the Data Encryption Standard", 2009 Available: http://www.cs.technion.ac.il/~biham/Reports/differential-cryptanalysis-of-the-data-encryption-standard-biham-shamir-authors-latex-version.pdf

[5] O. Dunkelman, G. Sekar, B. Preneel "Improved Meet-in-the-Middle Attacks on Reduced-Round DES", *Indocrypt 2007*, 15 p. Available: https://www.esat.kuleuven.be/cosic/publications/article-953.pdf

[6] Y. Baryshev "Models of pseudonondeterministic cryptography transformations" *Information technologies and computer engeneering*, Proceedings of the fifth international scientific-practical comference, Ivano-Frankivsk, 2015, pp. 189-191. (in Ukrainian)

[7] V. Luzhetskyi, Y. Baryshev "Data-driven pseudonondeterministic hashing constructions" *3rd International Scientific-Practical Conference Problems of Infocommunications Science and Technology*, *PICS and T 2016*. Available: https://ieeexplore.ieee.org/document/7905352/

[8] V. Luzhetskyi, Y. Baryshev "Pseudonondeterministic Hashing Conception" Systems of Control, Navigation and Communication, 3, 2010, pp. 94-98. (in Ukrainian)

[9] J. A Anderson. "Discrete mathematics with combinatorics", Prentice Hall, Upper Saddle River, New Jersey, 2004, 960 p.

[10] V. Glushkov "Digital automata synthesis", Moscow, Fizmathlit, 1962, 476 p. (in Russian)