UDC 004.421.5:004.056.55 **doi: 10.20998/2522-9052.2019.3.15**

V. Rudnytskyi[1], I. Opirskyy[2], O. Melnyk[3], M. Pustovit[3]

[1] Cherkasy State Technological University, Cherkasy, Ukraine
[2] Lviv Polytechnic National University, Lviv, Ukraine
[3] Cherkasy Institute of Fire Safety named after Chornobyl Heroes, Cherkasy, Ukraine

THE IMPLEMENTATION OF STRICT STABLE CRYPTOGRAPHIC CODING OPERATIONS

Abstract. According to the results of the study, it was established that the implementation of strict, stable cryptographic coding operations at the hardware and software levels does not cause difficulties, and in its simplicity and speed of implementation meets the requirements for implementation in stream encryption systems. Installed features and differences in various embodiments. In the hardware implementation, permutations will be performed between the bits to be processed, and in the software implementation, between the bits of the bytes of the same name that are processed. Regardless of the implementation option, according to the results of the operation, exactly half of the bits that took part in the conversion will be inverted, which ensures the maximum uncertainty of the encryption results.

Keywords: information security; cryptographic algorithm; cryptographic coding operations; strict robust coding.

Introduction

The global information networks' development and the implementation of computer and telecommunication technologies' achievements caused the creation of a single information space which is characterized both positive and negative consequences. Among the disadvantages of the single information space should be noted international cybercrimes and cyberwars.

The information protection in computer systems and networks requires a set of scientific as well as scientific and technical research implementation which should ensure the creation of a holistic system of organizational measures and application of specific information protection methods and means [1].

Cryptographic methods occupy a special place among the methods of information protection, because they, unlike other methods, rely only on the information properties itself and almost don't depend on the technologies of producing the information and telecommunication systems. The computer technology widespread use and the increase of the information volume cause a steady increase of interest in cryptography [2].

In order to guarantee a high degree of information security and to compensate the increased cybercriminals' opportunities through the development of computer technic and information technology, it is necessary to solve complex scientific and technical tasks on development and improvement of cryptography means constantly [3]. Based on this, the tasks of developing the cryptographic algorithms for information protection that provide maximum cryptographic stability have always been and will be relevant.

Literature review. The construction of cryptographic algorithms based on the operations of cryptographic information encoding (OCIE) use should be noted among the development directions of computer cryptography [4]. The synthesis of OCIE is based on the use of logical functions, the combination of which provides the substitution tables' simulation [5, 6]. One of the options to use these operations is their random generation based on the pseudorandom sequence [4]. One of the cryptographic algorithms' characteristics is the avalanche effect. The essence of the avalanche effect is in changing a small number of bits of the crucial sequence or input information that leads to a significant (avalanche) change in the number of bits of encrypted information [7]. The use of avalanche criterion and strict avalanche criterion (SAC) [8, 9] was proposed for quantifying the avalanche effect. The cryptographic algorithm satisfies the strict avalanche criterion, if in changing one bit of the input sequence, each bit of the output sequence changes with the probability of one half [9].

The conducted researches have shown that SAC does not allow to assess the suitability of the elementary function for implementation of OCIE, as well as to make it possible to evaluate the results of the implementation of the operations themselves [10]. The use of another criterion is proposed to evaluate the OCIE. The criterion of strict stable coding (SSC) is satisfied by the cryptographic algorithm or OCIE, if each bit of the output sequence changes relative to the input information with the probability of one half, regardless of the crucial sequence and the input information [11]. It has been established that only a small part of the OCIE corresponds to the SSC, and there are only four of them among the two-bit operations [12]. However, the research of the OCIE practical implementation has not received enough attention today.

The aim of the article is to investigate the implementation possibility of the operations of strict stable cryptographic coding at the hardware and software levels for their application in the systems of stream ciphering.

Exposition of basic material

We will restrict ourselves by the two-bit operations in holding the research of the OCIE implementation possibility, which meet the requirements of the SSC in the systems of stream ciphering.

A set of two-bit OCIE for SSC includes four operations [12], namely:

$$F_1 = \begin{bmatrix} x_1 \oplus 1 \\ x_2 \end{bmatrix}; \; F_2 = \begin{bmatrix} x_1 \\ x_2 \oplus 1 \end{bmatrix};$$

$$F_3 = \begin{bmatrix} x_2 \oplus 1 \\ x_1 \end{bmatrix}; \; F_4 = \begin{bmatrix} x_2 \\ x_1 \oplus 1 \end{bmatrix},$$

where $x_1, x_2 \in \{0, 1\}$ – the bits' value of the input information.

The data application of one-operand two-bit operations in streaming ciphering systems is realized on the basis of a random sample due to the use of a pseudorandom (subdued) sequence. As there are only four such operations, two bits are enough to ensure the choice of any of the operations. It should be mentioned that any selected operation is guaranteed to invert one of two bits of information, at the same time, F_2 invert the first and second bits respectively, and F_3 and F_4 with the probability of 0.5 invert the third or the fourth bits depending on the value of the input information [11]. A functional scheme of the device of selection and implementation the OCIE that meet the requirements of the SSC in the systems of stream ciphering is shown in Fig. 1.

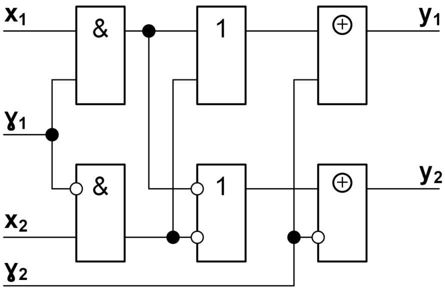

Fig. 1. A functional scheme of the device of selection and implementation the OCIE that meet the requirements of the SSC in the systems of stream ciphering

The functional scheme of the device, which was shown in Fig. 1, is described by the system of equations:

$$O^k = \begin{cases} \begin{bmatrix} x_1 \oplus 1 \\ x_2 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 0 \\ \begin{bmatrix} x_1 \\ x_2 \oplus 1 \end{bmatrix}, if\ \gamma_1 = 0;\ \gamma_2 = 1 \\ \begin{bmatrix} x_2 \oplus 1 \\ x_1 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 0 \\ \begin{bmatrix} x_2 \\ x_1 \oplus 1 \end{bmatrix}, if\ \gamma_1 = 1;\ \gamma_2 = 1 \end{cases} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \qquad (1)$$

where $x_1, x_2 \in \{0, 1\}$ – the bits' value of the input information, $\gamma_1, \gamma_2 \in \{0, 1\}$ – the bits' value of the subdued sequence.

The system of equations (1) could be regarded as a two-operand two-bit operation where the bits of encrypted input information are the first operand, and the bits of the subdued sequence are the second operand.

The practical application of the OCIE, which meet requirements of the SSC in the hardware implementation of the system of stream ciphering, does not cause complications, as it can be seen from the Fig. 1, which implements the operation model (1).

Let's consider the software implementation of the OCIE which are being researched. Nowadays, the vast majority of microcontrollers, microprocessors and processors of computer systems implement operations of informational bits processing in processing of at least a byte of information. This refinement makes the operation's (1) implementation cumbersome at the software level, and as a consequence, it is not fast enough and inefficient for stream ciphering.

In order to obtain the efficient use possibility of the OCIE that meet the requirements of the SSC in case of software implementation, it is to be constructed a two-operand operation of cryptographic transformation based on the model (1), similar to the operations of the modulo-2 addition operations with precision to the permutations, obtained in [13, 14]. To simplify the construction, let's represent this operation as a modulo-2 combination of the bits' permutation operations of the first operand and subduing the permutation results.

$$O^k = \begin{cases} \begin{bmatrix} x_1 \oplus 1 \\ x_2 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 0 \\ \begin{bmatrix} x_1 \\ x_2 \oplus 1 \end{bmatrix}, if\ \gamma_1 = 0;\ \gamma_2 = 1 \\ \begin{bmatrix} x_2 \oplus 1 \\ x_1 \end{bmatrix}, if\ \gamma_1 = 1;\ \gamma_2 = 0 \\ \begin{bmatrix} x_2 \\ x_1 \oplus 1 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 1 \end{cases} =$$

$$= \begin{cases} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 0 \\ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 1 \\ \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} \oplus \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 0 \\ \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} \oplus \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 1 \end{cases}.$$

$$O^k = O^{k*} \oplus \bar{O}^k =$$

$$= \begin{cases} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 0 \\ \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 1 \\ \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 0 \\ \begin{bmatrix} x_2 \\ x_1 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 1 \end{cases} \oplus \begin{cases} \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 0 \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; if\ \gamma_1 = 0;\ \gamma_2 = 1 \\ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 0 \\ \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; if\ \gamma_1 = 1;\ \gamma_2 = 1 \end{cases}.$$

Considering the bits of a subduing as bits of the second argument, we will construct a simplified operation O^{k*} without considering the bits' inversions.

$$O^{k*}=\begin{cases}\begin{bmatrix}x_1\\x_2\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=0\\ \begin{bmatrix}x_1\\x_2\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=1\\ \begin{bmatrix}x_2\\x_1\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=0\\ \begin{bmatrix}x_2\\x_1\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=1\end{cases}=$$

$$=\begin{cases}\begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\oplus x_2\cdot\gamma_1\\x_1\cdot\gamma_1\oplus x_2\cdot\overline{\gamma}_1\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=0\\ \begin{bmatrix}x_1\\x_2\end{bmatrix}=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\oplus x_2\cdot\gamma_1\\x_1\cdot\gamma_1\oplus x_2\cdot\overline{\gamma}_1\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=1\\ \begin{bmatrix}x_2\\x_1\end{bmatrix}=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\oplus x_2\cdot\gamma_1\\x_1\cdot\gamma_1\oplus x_2\cdot\overline{\gamma}_1\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=0\\ \begin{bmatrix}x_2\\x_1\end{bmatrix}=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\oplus x_2\cdot\gamma_1\\x_1\cdot\gamma_1\oplus x_2\cdot\overline{\gamma}_1\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=1\end{cases}= \quad (2)$$

$$=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\oplus x_2\cdot\gamma_1\\x_1\cdot\gamma_1\oplus x_2\cdot\overline{\gamma}_1\end{bmatrix}.$$

Let’s construct a two-operand operation for signal processing of inversion $\overline{O}^k$:

$$\overline{O}^k=\begin{cases}\begin{bmatrix}1\\0\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=0\\ \begin{bmatrix}0\\1\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=1\\ \begin{bmatrix}1\\0\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=0\\ \begin{bmatrix}0\\1\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=1\end{cases}= \quad (3)$$

$$=\begin{cases}\begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}\overline{\gamma}_2\\\gamma_2\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=0\\ \begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}\overline{\gamma}_2\\\gamma_2\end{bmatrix}, \ if\ \gamma_1=0;\ \gamma_2=1\\ \begin{bmatrix}1\\0\end{bmatrix}=\begin{bmatrix}\overline{\gamma}_2\\\gamma_2\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=0\\ \begin{bmatrix}0\\1\end{bmatrix}=\begin{bmatrix}\overline{\gamma}_2\\\gamma_2\end{bmatrix}, \ if\ \gamma_1=1;\ \gamma_2=1\end{cases}=\begin{bmatrix}\overline{\gamma}_2\\\gamma_2\end{bmatrix}.$$

Based on the modulo-2 addition, we will combine the models (2) and (3) and obtain a two-operand operation that implements the model (1):

$$O^k=O^{k*}\oplus\overline{O}^k=$$

$$=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\oplus x_2\cdot\gamma_1\\x_1\cdot\gamma_1\oplus x_2\cdot\overline{\gamma}_1\end{bmatrix}\oplus\begin{bmatrix}\overline{\gamma}_2\\\gamma_2\end{bmatrix}=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\oplus x_2\cdot\gamma_1\oplus\overline{\gamma}_2\\x_1\cdot\gamma_1\oplus x_2\cdot\overline{\gamma}_1\oplus\gamma_2\end{bmatrix}= \quad (4)$$

$$=\begin{bmatrix}x_1\cdot\overline{\gamma}_1\vee x_2\cdot\gamma_1\oplus\overline{\gamma}_2\\x_1\cdot\gamma_1\vee x_2\cdot\overline{\gamma}_1\oplus\gamma_2\end{bmatrix}=\begin{bmatrix}y_1\\y_2\end{bmatrix}.$$

The received two-operand operation’s implementation simplicity at the hardware level is not in doubt. Let's evaluate the effectiveness of its implementation at the software level, taking into account that instead of two-bit operands, we will use two-byte operands. If

$$x_1=\{x_{1.0},x_{1.1},x_{1.2},x_{1.3},x_{1.4},x_{1.5},x_{1.6},x_{1.7}\};$$

$$x_2=\{x_{2.0},x_{2.1},x_{2.2},x_{2.3},x_{2.4},x_{2.5},x_{2.6},x_{2.7}\};$$

$$\gamma_1=\{1,1,0,1,0,0,0,1\};\ \gamma_2=\{0,1,0,0,1,0,0,1\},$$

then in accordance with the model of operation (4) we will obtain:

$$y_1=\{\overline{x}_{2.0},x_{2.1},\overline{x}_{1.2},\overline{x}_{2.3},x_{1.4},\overline{x}_{1.5},\overline{x}_{1.6},x_{2.7}\};$$

$$y_2=\{x_{1.0},\overline{x}_{1.1},x_{2.2},x_{1.3},\overline{x}_{2.4},x_{2.5},x_{2.6},\overline{x}_{1.7}\}.$$

As it can be seen from the transformation results in the software implementation of operation (4), the bits’ permutation will be made not between the neighboring bits but between the same name bytes’ bits (words, double words, etc.), herewith, exactly a half of the bits of processed information will be inverted. It can be argued that the software implementation of two-bit OCIE that meet the requirements of the SSC based on the model (4) is not complicated, and, in its simplicity and implementation speed, meets the requirements for the implementation into the stream ciphering systems.

Conclusion

On the basis of the conducted research, it can be argued that the implementation of operations of strict stable cryptographic coding at the hardware and software levels is not complicated, but, in its simplicity and implementation speed, meets the requirements for the introduction into the stream ciphering systems. It should be mentioned that in the hardware implementation, the permutations will be performed between the bits that are being processed, and in the software implementation - between the same name bytes’ bits, which are being processed. However, despite this, according to the results of the operation execution, exactly a half of the bits that participated in the conversion will be inverted, and this fact provides the maximal uncertainty of the encryption results.

References

1. Venbo, Mao (2005), *Modern cryptography: theory and practice,* Publishing house «Vylyams», Moscow, 768 p.
2. Yakovlev, A.V., Bezbogov, A.A., Rodin, V.V. and Shamkin, V.N. (2006), *Cryptographic protection of information*, Publishing house TGTU, Tambov, 140 p.
3. Malec, I.O. (2011), “The role and problems of functioning of telecommunication systems in emergency situations”, *Bulletin of Lviv Polytechnic National University*, vol. 710: Computer Science and Information Technology, pp. 74–78.
4. Babenko, V.G., Rudnyckyj, S.V. (2012), “Realization of the method of information protection on the basis of matrix operations of cryptographic transformation”, *Information Processing Systems*, vol. 9 (107), pp. 130–139.
5. Rudnyckyj, V.M., Myronets, I.V. and Babenko, V.G. (2011), “Systematization of the Complete Set of Logical Functions for the Cryptographic Information Conversion”, *Information Processing Systems,* vol. 8 (98), pp. 184–188.

6. Rudnyckyj, V.M., Babenko, V.G. and Zhylyaev, D.A. (2011), “Algebraic structure of the set of logical coding operations”, *Science and Technology of the Air Forces of the Armed Forces of Ukraine*, vol. 2 (6), pp. 112–114.
7. Richard, A.M. (2005), *Codes: the guide to secrecy from ancient to modern times,* Chapman & Hall. CRC, P. 142.
8. Thomas, W. Cusick and Pantelimon, Stanica (2009), *Cryptographic Boolean Functions and Applications*, Academic Press, 248 p.
9. Rudnyckyj, V.M., Shuvalova, L.A. and Nesterenko, O.B. (2016), “Analysis of two-digit operations of cryptographic coding on the criterion of severe avalanche effect”, *Scientific works of the Petro Mohyla Black Sea State University,* vol. 271, pp. 74–77.
10. Rudnyckyj, V.M., Shuvalova, L.A. and Nesterenko, O.B. (2016), “Synthesis of operations of cryptographic transformation on the criterion of strictly stable coding”, *Bulletin of the Engineering Academy of Ukraine*, vol. 3, pp. 105–108.
11. Rudnyckyj, V.M., Shuvalova, L.A. and Nesterenko, O.B. (2017), “Method of synthesis of operations of cryptographic transformation on the criterion of strictly stable coding”, *Bulletin of the Cherkasy State Technological University*, vol. 1, pp. 5-10.
12. Babenko, V.G., Lada, N.V. (2014), “Synthesis and analysis of operations of cryptographic addition by module two”, *Information Processing Systems*, vol. 2 (118), pp. 116–118.
13. Babenko, V.G., Lada, N.V. (2016), “The technology of the study of operations by module two”, *Smart and Young,* vol. 11-12, pp. 49-54.

Received (Надійшла) 15.07.2019
Accepted for publication (Прийнята до друку) 21.08.2019

Відомості про авторів / About the Authors

Рудницький Володимир Миколайович – доктор технічних наук, професор, завідувач кафедри інформаційної безпеки та комп’ютерної інженерії, Черкаський державний технологічний університет, Черкаси, Україна;
Volodymyr Rudnytskyi – Doctor of Technical Sciences, Professor, Head of Information Security and Computer Engineering Department, Cherkasy State Technological University, Cherkasy, Ukraine;
e-mail: RVN_2008@ukr.net; ORCID ID: https://orcid.org/0000-0003-3473-7433

Опірський Іван Романович – доктор технічних наук, доцент, доцент кафедри захисту інформації, Національний університет «Львівська політехніка», Львів, Україна,
Ivan Opirskyy – Doctor of Technical Sciences, Associate Professor, Associate Professor of Information Protection Department, Lviv Polytechnic National University, Lviv, Ukraine;
e-mail: iopirsky@gmail.com; ORCID ID: https://orcid.org/0000-0002-8461-8996

Мельник Ольга Григорівна – кандидат технічних наук, старший науковий співробітник, доцент кафедри безпеки об’єктів будівництва та охорони праці, Черкаський інститут пожежної безпеки імені Героїв Чорнобиля, Черкаси, Україна;
Olga Melnyk – Candidate of Technical Sciences, Senior Research, Associate Professor of safety of construction and labor protection facilities Department, Cherkasy Institute of Fire Safety named after Chornobyl Heroes, Cherkasy, Ukraine;
e-mail: melnyk.olja.2014@gmail.com; ORCID ID: https://orcid.org/0000-0002-9671-108X

Пустовіт Михайло Олександрович – старший викладач кафедри техніки та засобів цивільного захисту Черкаського інституту пожежної безпеки імені Героїв Чорнобиля, Черкаси, Україна,
Mykhaylo Pustovit – Senior Instructor of safety of construction and labor protection facilities Department, Cherkasy Institute of Fire Safety named after Chornobyl Heroes, Cherkasy, Ukraine;
e-mail: m.pustovit@gmail.com; ORCID ID: https://orcid.org/0000-0001-5313-1459

Реалізація операцій строгого стійкого криптографічного кодування

В. М. Рудницький, І. Р. Опірський, О. Г. Мельник, М. О. Пустовіт

Анотація. За результатами дослідження встановлено, що реалізація операцій строгого стійкого криптографічного кодування на апаратному та програмному рівнях не викликає складності, а по своїй простоті та швидкості реалізації відповідає вимогам для впровадження в потокові системи шифрування. Встановлено особливості та відмінності при різних варіантах реалізації. При апаратній реалізації перестановки для отримання результату будуть проводитися між бітами, що обробляються, а в програмній реалізації – між однойменними бітами байтів, що обробляються. Незалежно від варіанту реалізації, за результатами виконання операції буде інвертовано рівно половину біт, які брали участь у перетворенні, що забезпечує максимальну невизначеність результатів шифрування.

Ключові слова: захист інформації; криптографічний алгоритм; операції криптографічного кодування; строге стійке кодування.

Реализация операций строгого устойчивого криптографического кодирования

В. Н. Рудницкий, И. Р. Опирский, О. Г. Мельник, М. А. Пустовит

По результатам исследования установлено, что реализация операций строгого устойчивого криптографического кодирования на аппаратном и программном уровнях не вызывает сложности, а по своей простоте и скорости реализации соответствует требованиям для внедрения в поточные системы шифрования. Установлены особенности и отличия при различных вариантах реализации. При аппаратной реализации перестановки для получения результата будут проводиться между обрабатываемыми битами, а в программной реализации - между одноименными битами байтов, которые обрабатываются. Независимо от варианта реализации, по результатам выполнения операции будет инвертировано ровно половину бит, которые принимали участие в преобразовании, что обеспечивает максимальную неопределенность результатов шифрования.

Ключевые слова: защита информации; криптографический алгоритм; операции криптографического кодирования; строгое устойчивое кодирование.