UDC 681.3.06

R.V. OLIYNYKOV, R.I. KIYANCHUK

Kharkiv National University of Radioelectronics, Kharkiv, Ukraine

PERSPECTIVE SYMMETRIC BLOCK CIPHER OPTIMIZED FOR HARDWARE IMPLEMENTATION

Confidentiality of data exchange in modern information and telecommunication systems is usually provided by application of symmetric block ciphers. At the same time widely used block ciphers are generally designed for software implementation (AES) or special-purpose hardware modules (DES, TripleDES). Their system-on-a-chip implementations with strict constraints to the number of logic gates and energy consumption are quite ineffective. Consequently, such systems require a new generation of cryptographic algorithms. Our paper examines requirements for block ciphers designed for lightweight hardware implementations, describes the perspective cipher specification, its properties and comparison with already existing ciphers.

Keywords*: lightweight cryptography, symmetric block cipher, hardware.*

Introduction

Ubiquitous computerization, mass deployment of pervasive devices in everyday life and extensive Internet access promises many benefits, but also causes significant risks related to confidential information processing. Financial applications, wireless sensor networks, RFID tags, electronic toll collection systems, all require secure data processing, ensuring its integrity and confidentiality [1].

Vast majority of modern symmetric block ciphers are designed for software implementation while the hardware implementations require intense computing resources (gates, area on a chip, frequency and energy consumption) for attaining acceptable efficiency. Harshly constrained pervasive devices capabilities do not allow the effective usage of existing reliable ciphers [2].

Thereby, the need for developing perspective symmetric block ciphers designed for effective hardware implementation and assuring moderate security level emerged.

One of the latest findings in this area is PRESENT cipher designed for hardware implementation on constrained devices.

1. PRESENT cipher description

The main goal when designing PRESENT was moderate security level, implementation efficiency and simplicity. It may be used on ultra constrained hardware when utilization of existing ciphers such as AES is impossible. The hardware PRESENT implementation requires only 1000 gate equivalents (GE) [3].

Cipher developments with the same targets in mind had already taken place earlier. HIGHT was published in 2006 [4]. It consists of a Feistel network and only 8-bit operations, has 64 bit input block size, 128 bit key length and 32 rounds. Its authors claim the hardware implementation to fit on 3048 gate equivalents.

mCrypton was published in 2006 and intended for both software and hardware implementations. It has 64-bit input block and consists of 13 rounds. The possible key length is 64, 96 or 128 bits. The hardware implementation of enciphering function requires at least 2420 gate equivalents.

Scalable Encryption Algorithm (SEA) was proposed in 2006 and targeted for constrained (software) devices with special emphasis on scalability [5]. Therefore SEA has a wide range of deployment. The input block size n, key length k, machine word b and number of rounds nr are variable cipher parameters.

The price for such scalability is implementation complexity that requires 3758 GE with $n = 96$ bit, $b = 8$ bit, $nr = 93$.

1.1. Perspective block cipher requirements for hardware implementation

PRESENT designers set the following requirements for the cipher [6].

- The cipher is to be implemented in hardware.
- Applications will only require moderate security levels.
- Applications are unlikely to require the encryption of large amounts of data. Implementations might therefore be optimised for performance or for space without too much practical impact.
- In some applications it is possible that the key will be fixed at the time of device manufacture. In such cases there would be no need to re-key a device (which

would incidentally rule out a range of key manipulation attacks).

- After security, the physical space required for an implementation will be the primary consideration. This is closely followed by peak and average power consumption, with the timing requirements being a third important metric.
- In applications that demand the most efficient use of space, the block cipher will often only be implemented as encryption-only. In this way it can be used within challenge-response authentication protocols and, with some careful state management, it could be used for both encryption and decryption of communications to and from the device by using the counter mode.

PRESENT is a symmetric block cipher with 64 bit input block size and 80 bit key length needed for assuring moderate security level. This is also the position taken for hardware profile stream ciphers submitted to eSTREAM project [7]. Specification also defines a 128 bit key. The PRESENT encipher and decipher hardware implementation still requires less space than AES encipher-only implementation [8]. The round keys can be computed on the fly during enciphering (each subkey per round, since round key computation consists in updating the key register).

PRESENT is a SPN and contains 31 rounds (fig. 1). The last 32-nd subkey is used for whitening after main loop. The loop consists of linear bit permutation layer and nonlinear substitution layer. The nonlinear layer uses 4-bit S-box which is applied 16 times to the whole input block each round. Key is injected into data via modulo 2 addition.

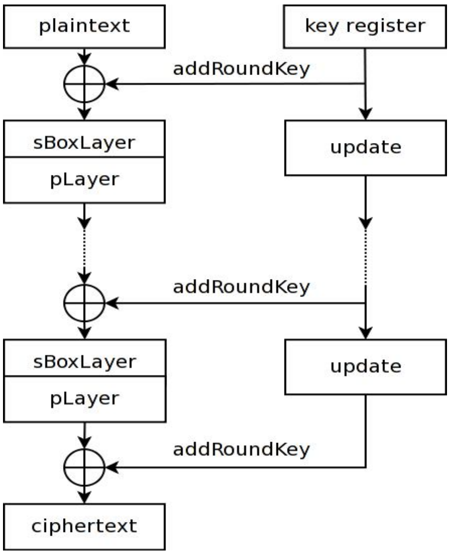

Fig. 1: PRESENT enciphering

1.2 Substitution layer

The nonlinear layer is represented by a single 4-bit substitution $F_2^4 \to F_2^4$. It is a direct outcome of strict constrains on efficiency and implementation area. Some additional restrictions are applied on S-box for reaching an avalanche effect. Let us denote the Fourier coefficient of S by

$$S_b^W(a) = \sum_{x \in F_2^4} (-1)^{(b,S(x))+(a,x)}. \qquad (1)$$

Than the PRESENT S-box (table 1) meets the following conditions:

1. for any fixed non-zero input difference $\Delta_I \in F_2^4$ and any fixed non-zero output difference $\Delta_O \in F_2^4$ it is required that
$$\#\left\{x \in F_2^4 | S(x) + S(x + \Delta_I) = \Delta_O\right\} \le 4; \qquad (2)$$
2. for any fixed non-zero input difference $\Delta_I \in F_2^4$ and any fixed output difference $\Delta_O \in F_2^4$ such that $wt(\Delta_I) = wt(\Delta_O) = 1$ we have
$$\#\left\{x \in F_2^4 | S(x) + S(x + \Delta_I) = \Delta_O\right\} = 0; \qquad (3)$$
3. for all non-zero $a \in F_2^4$ and all non-zero $b \in F_2^4$ it holds that $\left|S_b^W(a)\right| \le 8$;
4. for all $a \in F_2^4$ and all non-zero $b \in F_2^4$ such that $wt(a) = wt(b) = 1$ it holds that $\left|S_b^W(a)\right| \le 4$.

Table 1

PRESENT S-box

x	0	1	2	3	4	5	6	7
S[x]	C	5	6	B	9	0	A	D
x	8	9	A	B	C	D	E	F
S[x]	3	E	F	8	4	7	1	2

From entire set of S-boxes satisfying the specified conditions the one was chosen with the most efficient hardware implementation (that is less logic gates required for boolean representation). The boolean function of each S-box bit after minimization is showed below:

$$S_0(x) = x_3 \cdot x_2 \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} +$$
$$+\overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 + x_3 \cdot x_1 \cdot \overline{x_0} + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 +$$
$$+\overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot x_0 + x_3 \cdot \overline{x_2} \cdot \overline{x_0};$$
$$S_1(x) = \overline{x_3} \cdot x_2 \cdot x_1 \cdot \overline{x_0} + x_3 \cdot x_2 \cdot x_0 +$$
$$+\overline{x_2} \cdot x_1 \cdot \overline{x_0} +_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot x_0 +$$
$$+x_3 \cdot \overline{x_2} \cdot \overline{x_0};$$
$$S_2(x) = \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} + \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 +$$
$$+x_3 \cdot x_2 \cdot \overline{x_1} + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 + \overline{x_2} \cdot x_1 \cdot \overline{x_0} +$$
$$+x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0;$$

$$S_3(x) = \overline{x_3} \cdot x_2 \cdot x_1 \cdot \overline{x_0} + \overline{x_3} \cdot \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} +$$
$$+\overline{x_3} \cdot x_2 \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \cdot \overline{x_2} \cdot x_1 + \overline{x_3} \cdot x_2 \cdot x_1 \cdot x_0 +$$
$$+x_3 \cdot \overline{x_2} \cdot \overline{x_1} \cdot x_0 + \overline{x_3} \cdot \overline{x_2} \cdot x_1 \cdot x_0;$$

where $\overline{x_i}$ denotes the inversion of x_i bit, $\cdot$ denotes logical AND, $+$ denotes logical OR.

1.3. Permutation layer

The main problem considered during the permutation layer development was the number of required logic gates for implementation. Functional representation of permutation is showed in formula (4).

$$P(i) = \begin{cases} i \cdot 16 \bmod 63, & i \in 0,\ldots,62 \\ 63, & i = 63. \end{cases} \quad (4)$$

1.4. Key schedule

User key is stored in register K and is represented with a bit sequence $k_{79}k_{78}\ldots k_0$. Each round the subkey consists of 64 most significant (left) key register bits. After extracting another subkey the key register state is updated as follows (fig. 2):

1. $[k_{79}k_{78}k_1k_0] = [k_{18}k_{17}k_{20}k_{19}]$
2. $[K_{79}k_{78}k_{77}k_{76}] = S[k_{79}k_{78}k_{77}k_{76}]$
3. $[k_{19}k_{18}k_{17}k_{16}k_{15}] = [k_{19}k_{18}k_{17}k_{16}k_{15} \oplus \text{round\_counter}]$

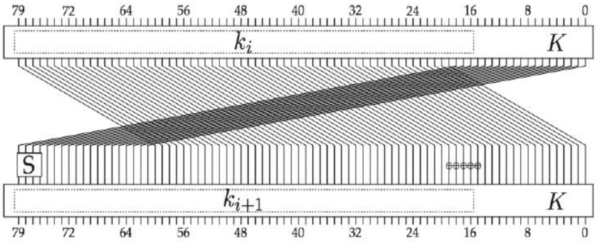

Fig. 2: The key schedule for PRESENT

1.5. Deciphering

Deciphering uses analogous inverse permutations. Subkeys are supplied in the same order as during enciphering.

The inverse S-box is showed in table 2.

Table 2

Inverse S-box of PRESENT

x	0	1	2	3	4	5	6	7
S[x]	5	E	F	8	C	1	2	D
x	8	9	A	B	C	D	E	F
S[x]	B	4	6	3	0	7	9	A

2. PRESENT hardware implementation properties

Cipher authors considered a wide variety of different target platforms ranging from highly-optimized ASICs, over more flexible but still efficient low-cost FPGAs to hardware-software co-design approaches and flexible software implementations for 4-, 8-, 16- and 32-bit processors.

We will investigate the ASIC implementation as it is the most efficient one.

2.1. Performance evaluation

To assess the efficiency of our implementation the developers used the following metrics.

Area: requirements are usually measured in μm^2, but this value depends on the fabrication technology and the standard cell library. In order to compare the area requirements independently it is common to state the area as gate equivalents [GE]. One GE is equivalent to the area which is required by the two-input NAND gate with the lowest driving strength of the appropriate technology. The area in GE is derived by dividing the area in μm^2 by the area of a two-input NAND gate.

Cycles: Number of clock cycles to compute and read out the result.

Time: The required amount of time for a certain operation can be calculated by dividing the amount of cycles by the operating $t = \dfrac{\text{cycles}}{\text{freq}}$.

Throughput: The rate at which new output is produced with respect to time. The number of output bits is divided by the time, i.e. by the needed cycles and multiplied by the operating frequency. It is expressed in bits per second [bps].

Power: The power consumption is estimated on the gate level by Synopsys PowerCompiler. It is provided in micro Watt $[\mu W]$.

Energy: The energy consumption denotes the power consumption over a certain time period. It can be calculated by multiplying the power consumption with the required time of the operation. The energy consumption is provided in micro Joule $[\mu J]$.

Current: The power consumption divided by the typical core voltage.

The area to throughput ratio is used as a measure of hardware efficiency. The hardware efficiency is calculated by dividing the area requirements by the throughput and is expressed in gate equivalents per bits per second $\left[\dfrac{GE}{bps}\right]$.

2.2. Measurements

Mearurements of ASIC round-based PRESENT implementation on 180 nm manufacturing technology with 4-bit datapath are done at 100 kHz frequency and presented below:

throughput: 11.7 Kbps;
area: 1075 GE;
efficiency: 10.89 bps/GE;
current: 2.78 μA .

Serialized and parallelized architectures implemention is also possible and described by authors in detail [3].

2.3. Cryptographic security

The substitution layer has been developed with the resistance to differential and linear cryptanalysis in mind. The maximum differential probability of a PRESENT S-box is 2^{-2} and so the probability of a single 25-round differential characteristic is bounded by 2^{-100}. The S-box has the following differential properties:

$\#0:159$
$\#2:72$
$\#4:24$

Linear properties of the S-box are showed below:

$\#0:123$
$\#2:96$
$\#4:36$

For approximating 28 cipher rounds with linear cryptanalysis one needs to obtain 2^{84} pairs plaintext/ciphertext, which exceeds the set of all possible plaintexts for PRESENT.

Integral attack on 7-round PRESENT requires $2^{43.3}$ choosen plaintexts, has time complexity $2^{100.1}$ and requires 2^{77} bytes of memory.

For applying an algebraic attack 11067 quadratic equations with 4216 variables have to be solved. Solving such equations is an NP-hard task. Despite the successful attacks on small-scale versions of block ciphers the increase of input block size results in enormous time and memory complexity.

The cipher security depends on key scheduling scheme. PRESENT uses round counter XORing with key register to decrease the correlation between subkeys. For the sake of nonlinearity while generating key material some bits pass through S-box when the key register updates (table 1).

All bits in the key register are a nonlinear function of the 80-bit master key by round 21. Each bit in the key register after round 21 depends on at least four of the master key bits

By deriving K_{32}, six bits are degree two expressions of the 80 master key bits, 24 bits are of degree three, while the remaining bits are degree six or degree nine function of the master key bits.

The statistical saturation attack can break 14 out of 31 rounds of PRESENT and requires 2^{34} plaintext/ciphertext pairs [9].

Side channel attacks as well as invasive attacks may be a threat for PRESENT just like for any other cryptographic primitives.

3. Comparison of PRESENT, GOST 28147-89 and AES

Advanced Encryption Standard (AES) is a symmetric block cipher, adopted as the national encryption standard in USA. This algorithm is widely used in the world and therefore very well researched.

GOST 28147-89 – symmetric block cipher, which is the encryption standard in Russia and other CIS countries, adopted in 1989. The most efficient known attack breaks the cipher only 2^8 times faster than a brute force. In 2010 the GOST cipher was submitted to ISO to become an international standard.

In consideration of long-term analysis and wide usage of AES and GOST 28147-89 the comparison of PRESENT with these ciphers is necessary.

3.1. Implementation complexity

The PRESENT structure is incredibly simple. After some precomputations the algorithm may be replaced with the lookup table and key addition. All cipher operations can be successfully performed on 4-bit processor and do not require complex calculations. The most compact implementation requires 1000 gate equivalents.

AES has complex structure [10]. It uses 8-bit S-box and its storage requires significantly more memory than 4-bit S-box. For linear bit diffusion the maximum distance separable (MDS) code based permutation is used. MDS-permutation uses matrix multiplications in $GF(2^8)$ and requires substantial computer resources or additional precomputed lookup tables. Hardware implementation fits on 3100 GE on 350 nm manufacturing process.

GOST 28147-89 represents a Feistel network. The cipher uses following operations: modulo 32 addition, bitwise exclusive OR, bitwise shift and a substitution box. Software implementation for 32-bit processors overtops PRESENT in throughput. The hardware implementation requires only 800 gate equivalents [11].

3.2. Cryptographic security

AES has 128-bit input block size and possible key length of 128, 192 or 256 bits. Number of rounds depends on the key length (10, 12 or 14). Best known attack on AES-256 has $2^{99.5}$ complexity but isn't applicable to AES-128. Side channel attacks on some implementations might have less complexity.

GOST 28147-89 has 64-bit input block size and the key length of 256 bits. Attack proposed by Nicolas Courtois requires 2^{64} plaintext/ciphertext pairs and speeds up cryptanalysis only by a factor of 2^{8} comparable to brute force [12]. It is worth noting, however, that any cipher can be broken by generating a dictionary of plaintext/ciphertext values for all possible inputs. Taking into account the efficiency of PRESENT and its 64-bit input block size it is possible to compute all plaintext/ciphertext pairs for foreseeable time using appropriate computing powers.

In distinction from the described ciphers PRESENT has 80 bit key length. The cipher is inapplicable for enciphering large data requiring high security level, but is well suited for embedded devices and RFID tags where the moderate security level for small data sequences is acceptable.

By comparison results GOST 28147-89 is also well suited for such applications and betters PRESENT on some parameters.

3.3. Efficiency

Ciphers efficiency comparison is showed in table 3. It is worth noting that the examined PRESENT implementation is designed for 4-bit processor (handles 4 bits per tick) whereas AES and GOST are quite ineffective to function on 4-bit processors.

Table 3

Efficiency comparison
of PRESENT, AES and GOST 28147-89

Cipher	Key	Block	T'put, Kb/s	Area, GE	Eff., bps/ GE
GOST	256	64	14	800	17,5
AES	128	128	80	3100	25,81
PRESENT	64	80	11,7	1075	10,89

Conclusion

PRESENT cipher was specifically designed for hardware implementation and functioning on ultra constrained devices. This fact explains design decisions towards simple and high-speed permutations and 4-bit S-box applied to the whole input block. The absence of computationally difficult operations (multiplication, modulo addition) and lookup tables ensure compact implementation, less area on a chip requirement and so -- cheap self-cost.

However the comparison for PRESENT, AES and GOST 28147-89 showed that GOST also fits for lightweight cryptography purposes and a little bit better than PRESENT in implementation area. With slight modification of the algorithm hardware implementation will require as little as 651 GE. Unlike new PRESENT cipher GOST 28147-89 is well time-tested and examined, also a wide variety of implementations already exists.

Taking into account the submitting of GOST cipher for international encryption standard, further research of the cipher applications on constrained devices (energy consumption, side channel attacks resistance) is reasonable.

In turn, PRESENT can function on 4-bit processors and has more flexibility for implementation that allows its effective use on devices with variate architectures.

References

1. Poschmann, A. Lightweight Cryptography From an Engineers Perspective [Text] / A. Poschmann. – Technical report, Horst-Görtz Institut für IT Sicherheit. – 2007.

2. Seys, S. Lightweight Cryptography Enabling Secure Wireless Networks. Workshop on Security Issues in Mobile and Wireless Heterogeneous Networks [Text] / S. Seys. – Katholieke Universiteit Leuven/Dept. Elektrotechniek (ESAT). Computer Security and Industrial Cryptography (COSIC). Brussels. – 2004.

3. Rolfes, C. Security for 1000 Gate Equivalents [Text] / C. Rolfes, A. Poschmann, C. Paar. – Ecrypt workshop SECSI - Secure Component and System Identification. – 2008.

4. HIGHT: A New Block Cipher Suitable for Low-Resource Device [Text] / H. Deukjo, S. Jaechul, H. Seokhie, et al. // CHES. – 2006. – P. 46 – 59.

5. SEA: A Scalable Encryption Algorithm for Small Embedded Applications [Text] / François-xavier st, G. Piret, N. Gershenfeld, J.J. Quisquater // Smart Card Research and Applications, Proceedings of CARDIS 2006, LNCS. Springer-Verlag. – 2006. – P. 222 – 236.

6. PRESENT: An Ultra-Lightweight Block Cipher [Text] / A. Bogdanov, C. Paar, A. Poschmann, et al. // Proceedings of CHES 2007. Springer-Verlag. – 2007.

7. The eSTREAM Project [Electronic resource]. – Access mode: http://www.ecrypt.eu.org/stream/. – 6.09.2011.

8. European Network of Excellence in Cryptology II [Electronic resource]. – Access mode: http://www.ecrypt.eu.org. – 2.08.2011.

9. Collard, B. A Statistical Saturation Attack against the Block Cipher PRESENT [Text] / B. Collard, F.X. Standaert // Proceedings of the The Cryptographers' Track at the RSA Conference 2009 on Topics in Cryptology in CT-RSA '09. Springer-Verlag. Berlin, Heidelberg. – 2009. – P. 195 – 210.

10. Daemen, J. The Design of Rijndael: AES - The Advanced Encryption Standard [Text] / J. Daemen, *V. Rijmen. – Springer Verlag, Berlin, Heidelberg, New York. – 2002.*

11. Poschmann, A. 256 bit standardized crypto for 650 GE: GOST revisited [Text] / A. Poschmann, S. Ling, H. Wang // Proceedings of the 12th international conference on Cryptographic hardware and embedded systems in CHES'10. Springer-Verlag. Berlin, Heidelberg, 2010. – P. 219 – 233.

12. Nicolas, T.C. Security Evaluation of GOST 28147-89 In View Of International Standardisation [Electronic resource] / T.C. Nicolas – Cryptology ePrint Archive, Report 2011/211, 2011. – Access mode: http://eprint.iacr.org. – 12.10.2011.

Надійшла до редакції 20.12.2011

Рецензент: д-р техн. наук, проф. І.Д. Горбенко, Харківський національний університет радіоелектроніки, Харків, Україна.

ПЕРСПЕКТИВНИЙ БЛОКОВИЙ СИМЕТРИЧНИЙ ШИФР, ОПТИМІЗОВАНИЙ ДЛЯ АПАРАТНОЇ РЕАЛІЗАЦІЇ

Р.В. Олійников, Р.І. Кіянчук

В сучасних інформаційно-телекомунікаційних системах конфіденційність инфомації, що передається, забезпечується, як правило, за допомогою блокових симетричних шифрів. В той же час блокові алгоритми, що використовуються, в основному орієнтовані або на програмну (AES) , або на спеціалізовані апаратні модулі (DES, TripleDES). Реалізація цих шифрів в системах-на-кристалах (system-on-chip) має жорсткі вимоги щодо кількості необхідних вентилів (gates) та низького енергоспоживання і є недостатньо ефективною. Відповідно, такі системи потребують нового покоління криптографічних алгоритмів. В роботі розглянуті вимоги до блокових симетричних шифрів, що призначені для компактної апаратної реалізації, наведено опис перспективного алгоритму, його властивості та порівняння з існуючими аналогами.

Ключові слова: компактна криптографія, блокові симетричні шифри, апаратне забезпечення.

ПЕРСПЕКТИВНЫЙ БЛОЧНЫЙ СИММЕТРИЧНЫЙ ШИФР, ОПТИМИЗИРОВАННЫЙ ДЛЯ АППАРАТНОЙ РЕАЛИЗАЦИИ

Р.В. Олейников, Р.И. Киянчук

В современных информационно-телекомуникационных системах конфиденциальность передаваемой информации, как правило, обеспечивается с помощью симметричных блочных шифров. В то же время широко используемые блочные алгоритмы, в основном ориентированы или на программную реализацию (AES), или на специализированные аппаратные модули (DES, TripleDES). Реализация этих шифров в системах-на-кристаллах (system-on-chip) с жёсткими требованиями к количеству необходимых вентилей (gates) и низкому энергопотреблению является достаточно неэффективной. Соответственно, такие системы требуют нового поколения криптографических алгоритмов. В работе рассмотрены требования к блочным шифрам, предназначенным для компактной аппаратной реализации, приведено описание перспективного алгоритма, его свойства и сравнение с существующими аналогами.

Ключевые слова: компактная криптография, блочные симметричные шифры, аппаратное обеспечение.

Олійников Роман Васильович – канд. техн. наук, доцент кафедри безпеки інформаційних технологій Харківського національного університету радіоелектроніки, Харків, Україна, e-mail: roliynykov@ gmail.com.

Кіянчук Руслан Ігорович – студент кафедри безпеки інформаційних технологій Харківського національного університету радіоелектроніки, Харків, Україна, e-mail: ruslan.kiyanchuk@gmail.com.