ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 6, Num. 1, 2021

IMPLEMENTATION KALYNA ALGORITHM IN MICROCONTROLLER

Volodymyr Bilenko, Valerii Hlukhov

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Authors' e-mail*: volodymyr.bilenko.ki.2017@lpnu.ua, glukhov@polynet.lviv.ua*

https://doi.org/10.23939/acps2021.01.___
Submitted on 01.06.2021

***Abstract*: The information security is playing an increasingly important role nowadays. Therefore, virus can be transmitted through the information in encrypted form. This is also applied to embedded systems. In this regard, the article is assigned to the topic of cryptocurrency protection in embedded systems using the national Ukrainian standard Kalyna. To further explore the topic, this algorithm was implemented on a microcontroller to test the performance, convenience and prospects for usage in embedded systems.**

***Index Terms*: encryption algorithm, symmetric block transformation, standard data encryption, cryptographic data protection, microcontroller.**

I. INTRODUCTION

The widespread use of computers and information technology is a very important and integral aspect of everyday life. With the proliferation of computer networks, the problem of secure information exchange is becoming more and more common. That is why the topic of information protection is now being updated. The field of cryptography deals with these issues. All modern standards for data encryption are developed in accordance with the requirements of the environment where they will be applied. Based on this, there is a pressing issue of analyzing the algorithms used to determine the stability, performance and security of the systems where they are used.

With the proliferation of embedded systems, the task of encrypting information at the hardware level is increasingly emerging. So-called cryptoprocessors are used for this type of task. They must meet specific requirements, depending on the environment where they are used.

The implementation of cryptographic algorithms and protocols in hardware requires complex controls, which makes logic vulnerable. In addition, hardware upgrades can be expensive and time consuming, which is clearly a disadvantage. But this does not preclude the fact that in today's world all data must be protected.

The most common solution is to use a general-purpose processor using one or more cryptographic coprocessors. This solution allows the implementation of sequential algorithms (which are often developed as a result of attacks and / or the evolution of standards) by a processor program, while tasks that can be performed in parallel are implemented in a coprocessor located inside the logical device. However, this solution causes some security issues: firstly, the general-purpose processor manipulates the keys as normal data, and modifying (intentionally or unintentionally) the contents of the program's memory can allow clear key reading outside the system; secondly, the use of general-purpose processors does not effectively isolate the red (unprotected) and black (protected) communication areas inside the device.

Currently, Ukraine uses algorithms for block symmetric data encryption DSTU GOST 28147: 2009, Triple DES, AES and DSTU 7624: 2014 [1], the implementation of latter is the purpose of this article.

II. PURPOSE OF THE ARTICLE

The purpose of the article is a comparative analysis of international data encryption algorithms and the national standard "Kalyna" taking into account the advantages and disadvantages of each of them. Also, there are additional objectives: to get acquainted with the features of microcontrollers to implement the cipher on them, to develop the structure of the algorithm on the target platform and approaches for the algorithm testing.

III. INFORMATION SECURITY AND PLACE FOR "KALYNA" ALGORITHM IN IT

As a result of today's global challenges to high levels of information security, the requirements for the security of information resources are growing steadily every year. Cryptographic security is the most reliable and effective method of information security, its main advantage is the protection of data without direct access to them. The main criterion for choosing a cryptosystem is data security, but not always. Also, the main role in cryptography is speed of data processing. Despite the wide range of modern methods and algorithms for encrypting information, not all of them have the required level of efficiency (optimality) under the conditions where system performance is being used [2]. As a result, old cryptographic algorithms are disappearing, and new ones are emerging and undergoing certain competitive practices to prove their ability to provide security over a period of time in the future.

Information security [3] is a set of tools and methods that provide information with the following basic properties:

- Privacy - data should not be available to third parties;
- Integrity - only a verified user has the right to modify the data;
- Availability - information can only be used by authorized users, according to certain requirements under which this information is distributed.

In accordance with the above properties, the following threats may arise in information security:

- Integrity (modification, destruction);
- Confidentiality (disclosure, leakage);
- Availability (unblocking);

Until 2014, there was an algorithm [4] in Ukraine, which was the national encryption standard. It did not meet modern speed requirements, so in 2015 it was replaced by a new standard - [1]. Although it has not been well studied, given the significant increase in security requirements, its implementation was necessary.

DSTU 7624: 2014 with the code name "Kalyna" [1] is a symmetrical block cipher for information protection. It was developed owing to the cooperation of Ukrainian scientists and the State Special Communications Service after the national competition in the field of cryptography. Kalina is a permutation-substitution network based on the AES cipher [5].

Code [4] provides:

• Extremely high stability

• High speed of implementations, both on lost and hardware platforms

• Comparable, and in some cases higher efficiency compared to the world's best solutions

• Availability of various modes of operation required for modern cryptosecurity

• Convenient implementation.

There are all possible combinations of the algorithm below (Table 1).

Table 1

Cipher states

Word size, bits	Block size, bits	Key size, bits	Identificator	Rounds
64	128	1*128=128	128/128	10
		2*128=256	128/256	14
	256	1*256=256	256/256	
		2*256=512	256/512	18
	512	1*512=512	512/512	

IV. "KALYNA" AND "AES" ALGORITHMS COMPARISION

Since the symmetric block encryption standards Kalyna and AES use similar cryptographic transformations, it will be expedient to compare these two algorithms [6].

The main differences between Kalyna and Rijndael (AES) are: increased resilience (number of encryption cycles);

the usage of addition modules 2 and 64 for the input of key information (allowing to protect against algebraic attacks, linear and differential cryptanalysis, interpolation attacks, etc.);

the usage of four S-blocks (blocks of nonlinear transformation) instead of one (additional protection against algebraic attacks, improvement of scattering properties of the algorithm - improved statistical properties, respectively, higher level of resistance to differential and linear cryptanalysis, etc.);

the usage of randomly generated four blocks selected by the criteria of resistance to differential, linear cryptanalysis, the degree of nonlinearity of Boolean functions (unlike the S-block Rijndael / Camellia and other ciphers using field calls and, accordingly, quadratic dependencies between input and output, - protection against algebraic attacks);

fundamentally new scheme of creating subkeys (protection against all known attacks on schemes of creating subkeys);

rather high productivity;

the ability to restore the session key on a separate subkey (additional protection against attacks that perform the recovery of subkeys).

All improvements are aimed at increasing resilience and preventing potential vulnerabilities in Rijndael [5] that have been identified in recent years.

V. MICROCONTROLLER ARCHITECTURAL FEATURES FOR "KALYNA" IMPLEMENTATION

For subsequent studies of the implementation of the cipher [1] one of the most common architectures of microcontrollers with 8/16/32-bit bit was chosen [10].

As an 8-bit platform, consider the family of AVR microcontrollers from Microchip (Atmel). This choice was made taking into account the successful command system of these MCs, which focuses on the maximum efficiency of programs written in high level language.

Among the features of the AVR core (Fig. 1. AVR central processor unit architecture), which are important in the field of cryptography for embedded systems, it should be noted that the memory has a Harvard architecture with an 8-bit SRAM data bus and 16-bit Flash memory. The register file includes 32 general-purpose registers that are directly connected to the ADC. All AVR family microcontrollers have a similar RISC core.

AVR microcontrollers support direct, indirect and direct addressing, given the availability of post-increment modes and pre-decrement ones, the ability to efficiently process data arrays during cryptographic algorithm execution, generating a compact program code. Indirect addressing is used to access data stored in Flash.

To test the Kalyna cipher on 16-bit platforms, Texas Instruments' MSP430 microcontroller was chosen as one of the most popular in its segment. One of the key benefits of the MSP430 family is the extremely low power consumption, which makes them very popular in embedded systems.

RISC-core controller MSP430 is built on the basis of Princeton architecture with a single space for addresses, commands and data and contains 16 registers, twelve of which (R4-R15) are general-purpose registers (**Ошибка! Источник ссылки не найден.**), and registers R0-R3 - perform special

functions . Compared to AVR, the set of commands here is very simple and consists of only 27 original and 24 emulated instructions, which are optimized for executing programs written in high-level languages. All commands are 16-bit, but they can handle both 16-bit and 8-bit operands. In total, support is available for seven addressing modes.

Non-volatile memory can be used for storing program code and data, which eliminates the need to copy data to RAM before next use. Due to the register operations performed in one clock cycle, orthogonal architecture guarantees compactness of the code and high performance. In the context of cryptography, an important feature of the processor is the exchange of data directly between memory cells.

The ARM Cortex processor from ARM was chosen as the platform for implementation on a 32-bit basis, because microcontrollers with ARM core make up at least 90% of the market of RISC microcontrollers with 32 bits and currently correspond to 8-bit models in terms of price and energy efficiency.

Given the topic of the article, we will consider the latest profile in the version of ARM Cortex-M.

ARM Cortex-M is a 32-bit processor based on the Harvard architecture with a three-level pipeline that implements a set of Thumb and Thumb-2 commands. The Cortex-M core (**Ошибка! Источник ссылки не найден.**) has 16 R0-R15 registers, of which R0-R12 are general purpose registers. The R13 register, which is a stack pointer, actually contains two more registers, but only one of them is available at a time. The primary pointer to the top of the MSP stack is used by the operating system kernel and interrupts handlers, and the pointer to the top of the PSP process stack is used by the application. Register R14 stores the return address when calling the function. The R15 register is a counter that contains the address of the command being executed. ADC has a 32-bit offset register, which allows in parallel with the operations to perform offsets of operands.

The kernel supports two levels of program code access (privileged and custom), which provide secure access to critical areas of memory, and has a basic model of the security mechanism. The address idle kernel is distributed fixedly.

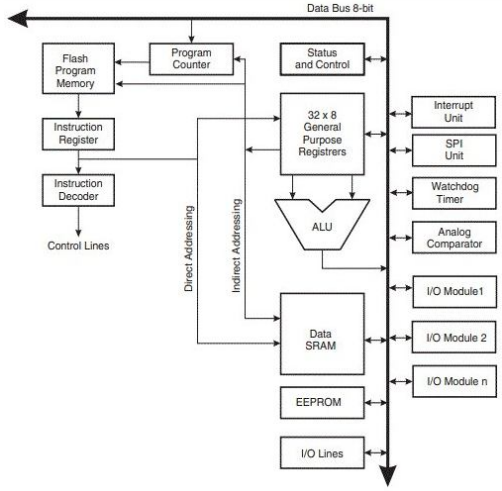

Fig. 1. AVR central processor unit architecture

VI. ALGORITHM STRUCTURE FOR MICROCONTROLLERS

The national encryption standard [1] refers to SPN, byte-oriented ciphers. The block size (len) and key length(key) are used to denote the cipher in the Kalyna (len, key) format [7].

When encrypting or decrypting, operations are performed on a two-dimensional byte array called the current state of the cipher (*State*). The current state of the cipher can be represented as a matrix of $8 \times c$ bytes (eight lines by c bytes): *State = (si,j)*, where $i = 0$ to 7, $j = 0$ to $c-1$.

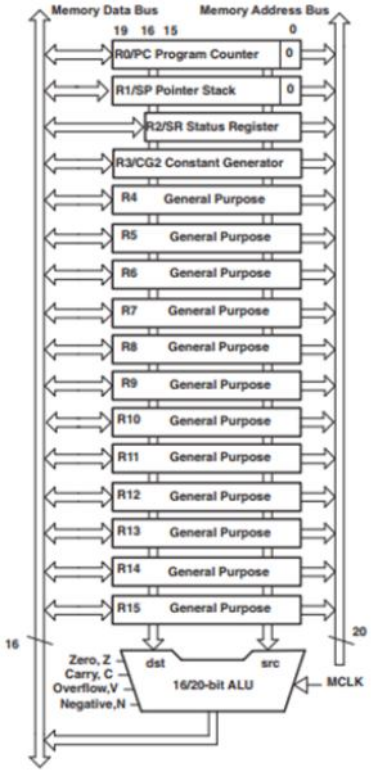

Fig. 2. MSP central processor unit architecture

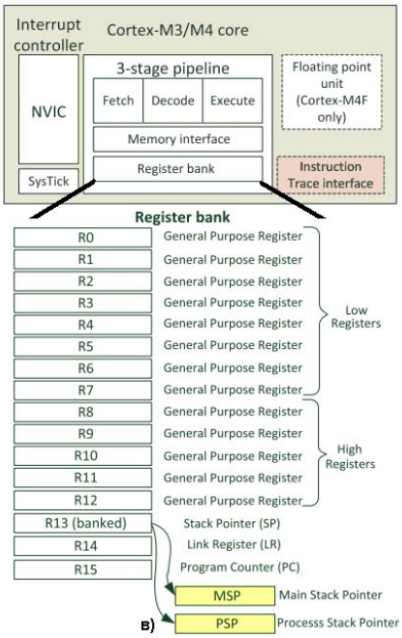

Fig. 3 ARM central processor unit architecture

The algorithm provides for the use of subtraction and addition operations modulo 264, addition modulo 2, linear transformation (*MixColumns, InvMixColumns*), table substitution (*SubBytes, InvSubBytes*) and cyclic row shift (*ShiftRows, InvShiftRows*). The structure of the algorithm "Kalyna" for the microcontroller is presented in (*Fig. 4* The structure of the microcontroller with the implementation of the algorithm).

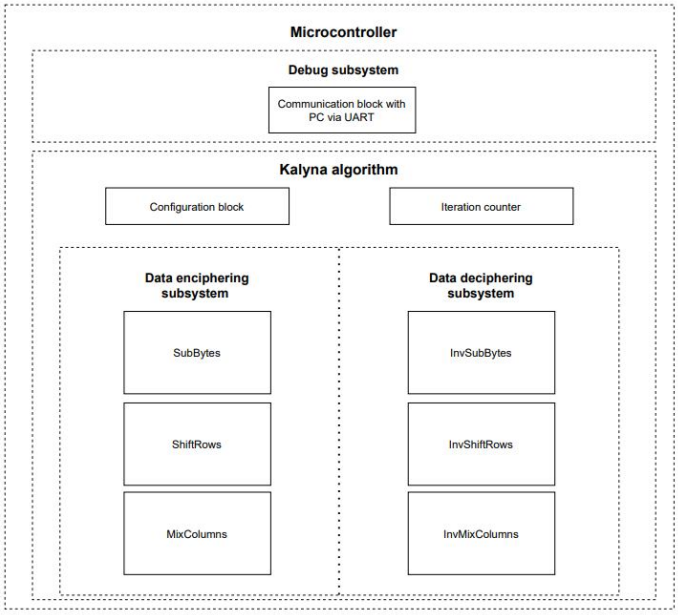

Fig. 4 The structure of the microcontroller with the implementation of the algorithm

Consider in more detail each of these operations [8].

Addition and *subtraction* operations - implement arithmetic addition or subtraction of columns of the state matrix and columns of the loop subkey modulo 2^{64}. The numbers in the columns are considered to be represented in the little-endian format, i.e., less significant bytes have a smaller index.

The operation of *adding modul*o 2 - performs the addition of modulo 2 state matrix *State* and cyclic subkey.

SubBytes and *InvSubBytes* operations - substitute each byte of the state matrix for its corresponding byte from one of the four replacement tables *S0-S3* and *inv S0-inv S3* for encryption and decryption operations, respectively. Each table is 256 bytes in size.

ShiftRows and *InvShiftRows* operations - cyclically shift the bytes of lines to the right or left, respectively. The number of positions δi to which the string is shifted depends on the line number i and the length of the block l and is calculated by the formula: $\delta i = (i * l) / 512$.

MixColumns and *InvMixColumns* operations - perform conversion of state matrix columns by performing multiplication and addition operations in a finite field *GF (28)* modulo irreducible polynomial $\psi = x8 + x4 + x3 + x2 + 1$. Each element of the resulting of the state matrix $W = (wi, j)$ is calculated in the field GF (2^8) as a scalar product of the row of the matrix v (*inv_v*) on the column of the *State* matrix of the state according to the formulas:

$w_{i,j} = (v >>> i) \otimes Sj,$

$w_{i,j} = (inv\_ v <<< i) \otimes Sj,$

where $v = (01h, 01h, 05h, 01h, 08h, 06h, 07h, 04h)$, $inv\ v = (ADh, 95h, 76h, A8h, 2Fh, 49h, D7h, CAh)$; *Sj* - j-th column of the state matrix; $v >>> i$ and $v <<< i$ - operations of cyclic shift of bytes of vector v to the right and to the left on and positions accordingly.

To obtain round subkeys from the original master key, the key deployment procedure is used, which involves the standard transformations discussed above.

VII. TESTING PRINCIPALS

Since the main requirements for cryptoalgorithm in devices for different purposes can be both economical memory usage and high speed, it is advisable to study the algorithms from this point of view [3]. To achieve maximum program speed, you should optimize those parts of the algorithm that are the most complex in terms of computing, using commands and addressing methods that require a minimum number of clock cycles MK. Since the operations of addition and summation modulo two are essentially atomic and are implemented using the appropriate processor instructions, the speed of the algorithms will generally be determined by the speed of linear transformation.

To test the algorithm, follow these steps:

• make sure that the main functions are performed correctly with the help of test kits

• check the correct implementation of the application mode **Ошибка! Источник ссылки не найден.** defined in the standard: gamma mode (CTR), encrypted text feedback (CFB), simulation input (CMAC); encrypted text communication (CBC); reverse braking gamma code (OFB); selective accelerator production with accelerated simulation insert (GCM, GMAC); manufacturing simulation scaling inserts (CCM); indexed replacement (XTS); Key data protection (KW) using test scenarios.

The test should be performed sequentially - first the basic procedures, and then the modes of application of the algorithm. Implementation is considered verified if the basic procedures and the basic modes of operation are performed.

In the process of research [9][7] of this topic, test scenarios were developed for the deployment of a key in a cyclic key for encryption, decryption and simulation functions. Usage of debuggers and specialized communication interfaces are the main features of testing the algorithm on a microcontroller.

VIII. VERIFICATION

Finally, the main part of the article is being tested. Platform for testing (Fig. 5 Functional scheme of microcontroller cel) is based on 32 bit ARM core MCU – STM32G071RBT. As it can be seen from the diagram below, the system provides communication with a personal computer by converting USB signals into UART telegrams. The firmware will be loaded from the KEIL uVision IDE using the official ST-Link V2 debugger from STMicroelectronics. Power supply and reset units will be provided also.

Procedures were performed for Kalyna (128,128) – 10 rounds and Kalyna (128, 256) – 14 rounds with implemented MDS profile which uses precalculated table for multiplication

and nonlinear substitution. This version focuses on maximum performance due to significant increase in code size. The essence of this technique is to combine operations SubBytes, ShiftRows and MixColumns into one using pre-calculated tables.

For data exchange USB to UART converter PL2303HX [10] with build in baud rate generator which supports speed up to 1.2M bps. In our case UART part has the following setup (**Ошибка! Источник ссылки не найден.**).

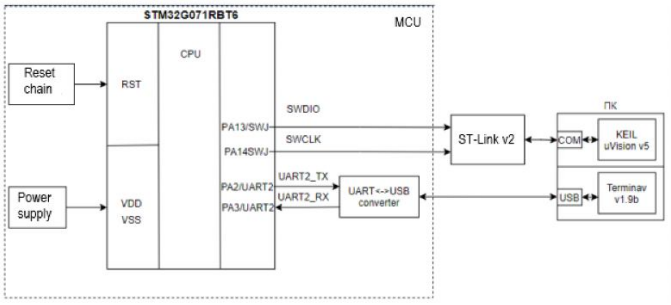

Fig. 5 Functional scheme of microcontroller cell

Table 2

UART setup

Baud rate	Data field	Stop bits	Parity
115200b	8 bits	1	none

As for USB part, it works as a bridge between main USB port and RS232 serial port. The two large on-chip buffers accommodate data flow from two different buses. The USB bulk-type data is adopted for maximum data transfer. And it is fully compatible with USB Specification v1.1. [11].

Finally, we got the cryptoprocessor cell with the next sizes (*Fig. 6*). and electrical characteristics.

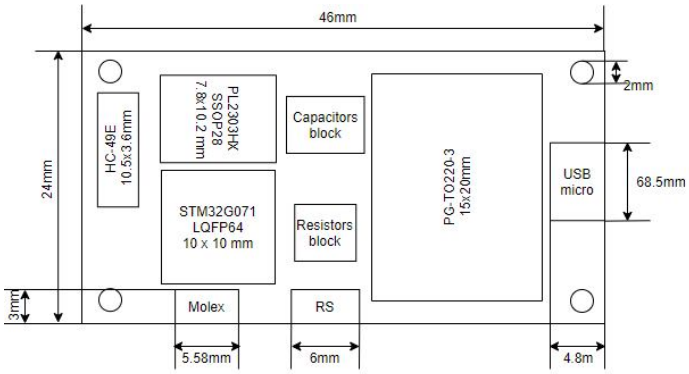

Fig. 6 Cryptoprocessor cell

Table 3

Electrical characteristics

Voltage	5V
Power consumption	~290mW (when 50°C)
Operating temperature	-40 to +85°C

The next step is to compile source code and find out the size of executable. From the (*Fig. 7* Program compilation) it could be seen that hex file size – 16640 bytes.

Load .hex file into MCU and check program correct execution (*Fig. 8*Fig. 8). Test scenario has been taken from [1]. If to compare our results with those pre-calculated from DSTU, it can be sure that it works appropriately.

```
..\control\main.c: 2 warnings, 0 errors
linking...
Program Size: Code=16640 RO-data=356 RW-data=2224 ZI-data=9152
FromELF: creating hex file...
"..\..\__build\kalyna.axf" - 0 Error(s), 2 Warning(s).
Build Time Elapsed:  00:00:02
```

Fig. 7 Program compilation

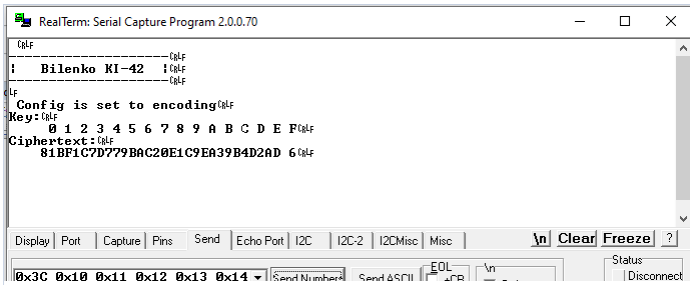

Fig. 8 Program execution

The last step is to measure time for transformations in milliseconds (ms). Input data is 800 bytes (50 blocks) packages. Based on mentioned data, convert it to speed in bytes per second (bps). Results are given in the table below (Table 4Table 4). Then convert measured data in time for transformations for one block (16 bytes), see (Table 5).

Table 4

Kalyna transformation speed

Mode	Data size, blocks	Encipher,		Decipher,	
		ms	bps	ms	bps
Kalyna MDS(128,128)	50	248	3225	264	3030
Kalyna MDS(128,256)	50	297	2693	314	2547

Table 5

Enciphering and deciphering speed for 1 block

Mode	Encipher time, ms	Decipher time, ms
Kalyna (128, 128)	4,96	5,28
Kalyna (128, 256)	5,94	6,38

Finally, compare execution speed with other widely used platforms such as PC with Interl Core I5 6300HQ CPU and FPGA Xilinx Spartan 6. All data have been taken from [13]. The following table (Table 6) demonstrates transformation speed in bytes per second (bps).

Table 6

Kalyna speed comparison at different platforms, bps

Platform	Core frequency	Kalyna (128, 128)		Kalyna (128, 256)	
		Enciph.	Deciph.	Enciph.	Deciph
PC	3.4 GHz	326 M	304 M	226 M	203 M
FPGA	375 MHz	22675	21088	18743	17727
MCU	64 MHz	3225	3030	2693	2547

As it is seen from the table, MCU has the worst performance, but, in counterweight to this disadvantage, there

are features that help to find out the place in embedded systems due to fast growth of that field. In accordance with that, we have to single out the following advantages:

- Low cost of components
- Faster and chipper development process
- Easy debugging
- Easy integration
- The lowest power consumption
- Small sizes

Due to the mentioned features, microcontrollers become more popular and can compete with traditional software and hardware implementation methods.

IX. CONCLUSION

A comparative analysis of the AES algorithm and the Ukrainian national standard Kalyna was conducted assessing the advantages and disadvantages for each of them. The acquaintance with the features of microcontrollers, which are suitable for the implementation of the Kalyna algorithm, was carried out. The structure of the algorithm for the target platform and approaches for testing were developed. Algorithm speed execution was analyzed for Kalyna (128,128) and Kalyna (128,256) with MDS profile. Speed comparison for different platforms was performed.

The development results using STM32G071RBT MCU are the following:

- Cryptoprocessor's cell sizes – 24mm x 46mm
- Voltage – 5V
- Power consumption - 290mW
- Size of executable file - 16640 bytes
- Kalyna (128, 128) enciphering speed – 3225 bytes/s
- Kalyna (128, 128) deciphering speed – 3030 bytes/s
- Kalyna (128, 256) enciphering speed – 2693 bytes/s
- Kalyna (128, 256) deciphering speed – 2547 bytes/s

References

[1] DSTU 7624: 2014 (2015). Information Technology. Cryptographic information protection. Symmetric block transformation algorithm. Kyiv, Ukraine: Ministry of Economic Development of Ukraine, p.228.

[2] Kuznetsov, O. O., Oliynikov, R. V., Gorbenko, Y. I., Pushkarev, A. I., Dirda, O. V., Gorbenko, D. I. (2015) Requirements justifications construction and analysis of perspective symmetric cryptographical transformations on the base of block bipher codes. Computer systems and networks, 806, pp.124-141.

[3] Karachka, A. F. (2017). Technologies of information security, Ternopil, Ukraine: Ternopil national economic university, 86 p.

[4] DSTU GOST 28147 (2009). Information processing systems. Cryptographic security. Cryptographic transformation algorithms. Kyiv, Ukraine: National standard of Ukraine, 28 p.

[5] FIPS-197: Advanced Encryption Standard (AES) (2001). Federal Information Processing Standard, National Institute of Standards and Technology, U.S. Dept. of Commerce, 47 p.

[6] Efimenko, A. A., Bailyuk, E. M., Pokotylo, O. A. (2018). Comparative analysis of the algorithm of symmetric block transformation Kalyna with other international standards for data encryption. Collection of scientific works of the Zhytomyr Military Institute, 18, pp. 124-142.

[7] Oliynikov, R., Gorbenko, I., Kazimirov, O. (2015). Principles of construction and basic properties of the new national standard of block encryption of Ukraine. Information security: Sec. Mag., 17(2), April – June 2015, pp. 142-157.

[8] Sovin, I. R., Otenko, V. I., Stefanyuk, E. F. (2017). Effective implementation of the block symmetric encryption algorithm DSTU 7624: 2014 for 8/16/32 bit embedded. Kyiv, Ukraine: Modern information security, pp. 6-16

[9] Dolgov, V. I., Oleinikov, R. V., Bolshakov, A. Y., Grigoriev A. V., Drobotko E. V. (2010). Cryptographic properties of reduced version of «Kalyna» cipher. Applied Radio Electronics, 9(3), pp. 349-354

[10] Domina, M., Bilenko, V., Hlukhov, V. (2019) Validation of Implementation Kalyna Block Cipher with The Help of Test Examples. In: International forum «Litteris et Artibus». Lviv, Ukraine: Lviv Polytechnic National University, pp. 62-64.

[11] Profilic Technology Inc. (2005). PL-2303 Edition USB to Serial Bridge Controller Product Datasheet. [online] Available at: https://www.estudioelectronica.com/wp-content/uploads/2018/10/PL2303.pdf [Accessed 30 May 2021]

[12] Intel, Compaq, Microsoft, NEC (1998). Universal serial bus specification. [online] Available at: https://composter.com.ua/documents/usb-rev1.1.pdf [Accessed 30 May 2021]

[13] Gorbenko, I., Halimov, H., Lisitskaya, I., Dolgov, V., Horbenko, Y., Ruzhencev, V., Vynokurova, E., Oleinikov, R. (2014). States analysis, development directions determination, standartization, improvement, development and implementation of cryptographic systems, including the EDS systems. Kharkiv, Ukraine: "Kharkiv national university of radioelectronics", 374 p.

Volodymyr Bilenko is a fourth-year student of Computer Engineering Department at Lviv Polytechnic National University. He was involved in the development of systems for military purposes with encryption topics. His is interested in topics related to embedded system engineering such as Internet of Things (IoT), Robotics and Applied Automation Systems.

Valerii S. Hlukhov is a professor of the Department of Computer Engineering Department at Lviv Polytechnic National University, Ukraine. He graduated from Lviv Polytechnic Institute with the engineer degree in Computer Engineering in 1977. In 1991 he obtained his Ph.D. from the Institute of Modeling Problems in Power Engineering of the National Academy of Science of Ukraine. He was recognized for his outstanding contributions into special-purpose computer systems design as a Senior Scientific Researcher in 1995.

He was awarded the academic degrees of Doctor of Technical Sciences in 2013 at Lviv Polytechnic National University. He became a Professor of Computer Engineering in 2014. He has scientific, academic and hands-on experience in the field of computer systems research and design, proven contribution into IP Cores design methodology and high-performance reconfigurable computer systems design methodology. He is an experienced reseracher in computer data protection, including cryptographic algorithms, cryptographic processors design and implementation. Prof. Hlukhov is an author of more than 100 scientific papers, patents and monographs.