Development of a New Lightweight Encryption Algorithm

Ardabek Khompysh[1], Nursulu Kapalova[2], Oleg Lizunov[3*], Dilmukhanbet Dyusenbayev[4], Kairat Sakan[5]
Information Security Laboratory, Institute of Information and Computational Technologies, Almaty, Kazakhstan[1, 2, 3, 4, 5]
Department of Information Technology, Al-Farabi Kazakh National University, Almaty, Kazakhstan[1, 5]

***Abstract*—Due to the growing need to use devices with low hardware resources in everyday life, the likelihood of their susceptibility to various cyber-attacks increases. In this regard, one of the methods to ensure the security of information circulating in these devices is encryption. For devices with small hardware resources, the most applicable is low-resource (lightweight) cryptography. This article introduces a new lightweight encryption algorithm, ISL-LWS (Information Security Laboratory – lightweight system), designed to protect data on resource-constrained devices. The encryption algorithm is implemented in the C++ programming language. The paper presents the statistical properties of ciphertexts obtained using the developed algorithm. For the experimental testing for statistical security, the sets of statistical tests by NIST and D. Knuth were used. Separately, the ISL-LWS algorithm was tested for avalanche effect properties. The obtained results of statistical tests were compared with the Present and Speck modern lightweight algorithms. The study and comparative analysis of the speed of encryption and key generation of the three algorithms were carried out on the Arduino Uno R3 board.**

Keywords—Lightweight block cipher; S-box; linear transformation; avalanche effect; IoT devices; RFID tags; null hypothesis; NIST tests; D. knuth tests

I. Introduction

The main directions of the development of cryptography are largely associated with the development of communications and information technology. It is the progress in these areas that has made possible the widespread use of compact devices with low computing power that have access to the Internet and implement the concept of the Internet of Things (IoT) [1][2]. Examples of such devices are radio frequency tags (RFID), automated process control systems (SCADA), wireless sensors, electronic personal identification tools, etc. [3].

Lightweight ciphers are often less secure than traditional ciphers such as AES. This is because lightweight ciphers are optimized for high speed and low power consumption, not maximum security.

As defined by the US National Institute of Standards and Technology (NIST), lightweight cryptography is a sub-category of cryptography that aims to provide solutions for high-growth applications that make extensive use of low-power smart devices [4][5]. Modern cryptographic algorithms can work well on computers, servers, and some mobile phones, but IoT devices, smart cards, and RFID tags require the use of lightweight cryptographic algorithms [6].

When building lightweight block encryption algorithms, the following architectural solutions are used [7]:

- Reduction of the block size from 128 bits to 64 bits;
- Use of keys 64, 80, and 128 bits long;
- Use of 4-bit S-boxes instead of 8-bit ones;
- Use of a simplified key schedule.

Designing algorithms based on well-studied and widely used operations that perform elementary linear/nonlinear transformations.

When creating lightweight block ciphers, the following structures are used [8]:

- Feistel network;
- Substitution-permutation network (SP-network) using substitution boxes of small length;
- LRX-structure (logical operations, rotate left (right) shift, and addition modulo 2);
- ARX-structure (addition modulo 2^n, rotate left (right) shift, and addition modulo 2).

One of the main issues in lightweight cryptography is achieving a balance between security, efficiency, and cost. Obviously, optimizing a lightweight cipher to achieve high speed can weaken some of its security properties, and the algorithm will be more vulnerable to some attacks. Therefore, when developing a lightweight cipher, the first step is to determine the requirements for its security and limited resources, taking into account the scope of its application. When developing the encryption algorithm, the authors tried to balance security and speed.

This article presents a new lightweight symmetric block cipher algorithm ISL-LWS and its statistical analysis. The scientific novelty of the proposed algorithm is the SP transformation, which is performed in parallel by linear (P-box) and non-linear (S-box) cryptographic primitives, where two S-boxes are used simultaneously. This procedure makes it possible to increase the degree of non-linearity and data confusion in fewer rounds. An overview of related work is presented in the next Section II. Section III presents the developed algorithm, which is designed according to the Feistel network and includes linear and non-linear transformations that provide a high level of diffusion and confusion. The round key schedule algorithm is also presented here. The results and discussion of the statistical tests are presented in Section IV. In addition, this section describes data on the hardware-software implementation of the algorithm and comparative performance analysis. Section V presents the conclusion, where the results of the work are indicated.

II. Related Work

To date, a fairly large number of lightweight block encryption algorithms based on SP networks and Feistel networks are known [9]. Both approaches have their advantages and disadvantages in the context of constructing algorithms in conditions of limited resources. Lightweight block ciphers are represented by the following algorithms: Present [10][11], Clefia [12], Katan [13], Simon [14], Speck [15], Secure IoT (SIT) [16], etc.

A study by Xinxin Fan et al. (Fan et al. 2013) introduced a lightweight WG-8 encryption algorithm of the Welch-Gong family of stream ciphers, adapted for devices with low hardware resources [17]. Typically, some of them have been improved and developed by simplifying block ciphers to improve their performance. For example, DESL which is also known as lightweight DES, is a variant of classic DES. The main difference between the DESL cipher and the DES algorithm is that the former uses one S-box instead of eight ones, which reduces the ROM requirements for storing tables by eight times.

The lightweight encryption algorithm Present [18] is described in the article by L.K. Babenko, D.A. Bespalov, O.B. Makarevich, R.D. Chesnokov, and Ya.A. Trubnikov. The authors of this article have developed a software implementation and synthesized it into a hardware unit for a system on a chip within the framework of the requirements for low-resource cryptography, having obtained a sufficiently effective solution for its application in devices. In 2012, the ISO and IEC organizations included the Present algorithm in the international standard for lightweight encryption ISO/IEC 29192-2:2012.

Speck is a block lightweight encryption algorithm developed by the US National Security Agency. Speck is one of the fastest in lightweight cipher benchmarks, but its performance is highly dependent on architecture. Speck supports several block and key sizes. The block length can be 32, 48, 64, 96, and 128 bits. The key length depends on the block size. The range of key sizes is 64, 72, 96, 128, 144, 192, and 256 bits. The number of encryption rounds depends on the block size and the key. The range of rounds is 22, 23, 26, 27, 28, 29, 32, 33, and 34. Speck is standardized by ISO within the RFID air interface standard [15].

In a study by Muhammad Usman et al. 64-bit block lightweight encryption algorithm SIT [16] with a key length of 64 bits is considered. The architecture of the algorithm is a mixture of a Feistel network and an SP network. Conducted studies show that the algorithm provides significant security after five rounds of encryption.

Thus, R&D on the development and study of lightweight encryption algorithms is relevant.

III. Lightweight Encryption Algorithm ISL-LWC

The block diagram of the proposed ISL-LWC lightweight block encryption algorithm is shown in Fig. 1.

The main parameters of the algorithm:

- block length – 64 bits;
- key length – 80 bits;
- number of encryption rounds - 16.

The algorithm uses SP transformation, modulo 2 addition (XOR operation), rotate shift, and non-linear transformations in the form of S-boxes (S).

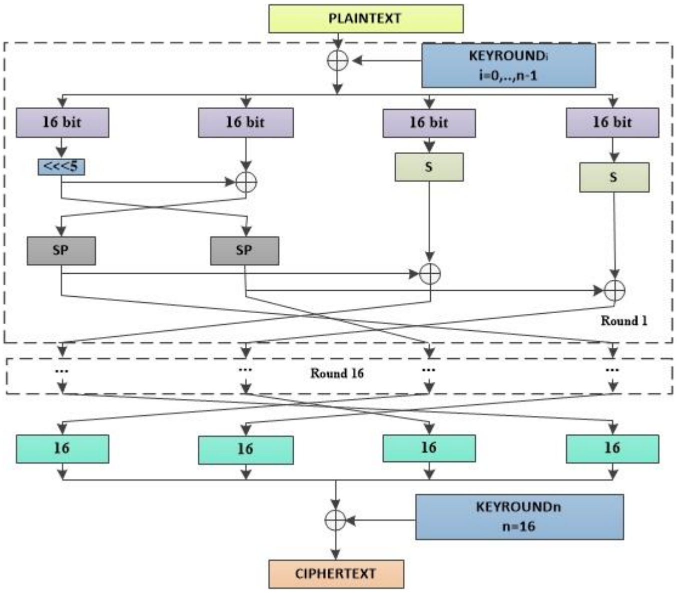


Fig. 1 Scheme of the encryption algorithm.

The encryption process consists of 4 stages:

Stage 1. A 64-bit plaintext block is added to the round key modulo 2 (XOR operation). Next, the resulting 64-bit block is divided into 4 subblocks of 16 bits each (the subblocks are numbered from left to right).

Stage 2. The 1st input subblock is rotated by 5, then the obtained value of the 1st input block is summed (XOR operation) with the 2nd subblock, and the resulting values are swapped in accordance with the scheme and go through SP transformations.

Stage 3. The 3rd and 4th sub-blocks go through the transformation S and then are added (XOR operation) with the results obtained at Stage 2 according to the scheme.

Stage 4. The results of Stages 2 and 3 are swapped according to the scheme of the encryption algorithm.

1) SP transformation: The SP transformation (Fig. 2) consists of non-linear 4-bit substitutions S-box1 and S-box2 (Tables I, II) and a linear bit permutation P-box (Table III). The methods for obtaining S-box1 and S-box2 are shown in [19]. The transformations above make it possible to perform confusion and diffusion.

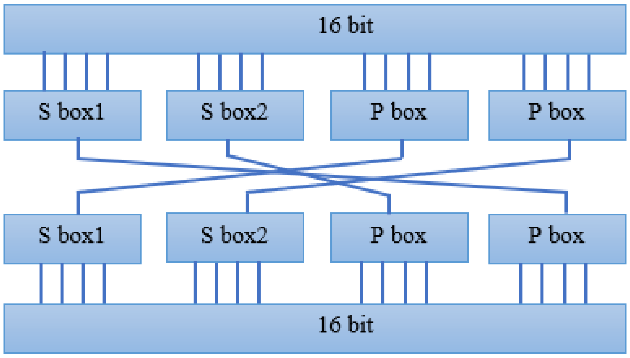


Fig. 2 SP transformation scheme.

TABLE I S-BOX1 SUBSTITUTION

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

TABLE II S-BOX2 SUBSTITUTION

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

TABLE III P-BOX BIT PERMUTATION

i	0	1	2	3
P(i)	3	2	0	1

2) S transformation; Input 16 bits are represented as $a_0a_1a_2a_3a_4a_5a_6a_7a_8a_9a_{10}a_{11}a_{12}a_{13}a_{14}a_{15}$ of which every sequential 4 bits are represented as m_i, $i = \overline{0,3}$. $m_0 = a_0a_1a_2a_3$, $m_1 = a_4a_5a_6a_7$, $m_2 = a_8a_9a_{10}a_{11}$, $m_3 = a_{12}a_{13}a_{14}a_{15}$. The values m_0, m_2 and m_1, m_3 are passed through 4-bit S-box1 and S-box2 and then swapped according to Fig. 3.

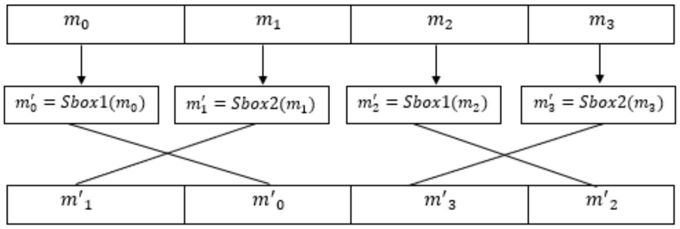


Fig. 3 S-box transformation process.

Round subkeys are generated on the basis of an 80-bit base key, which is divided into five sub-blocks of 16 bits each (sub-blocks are numbered from left to right) (Fig. 4). The cryptographic transformations used are the 4-bit S-box and addition modulo 2 raised to the power of the word length.

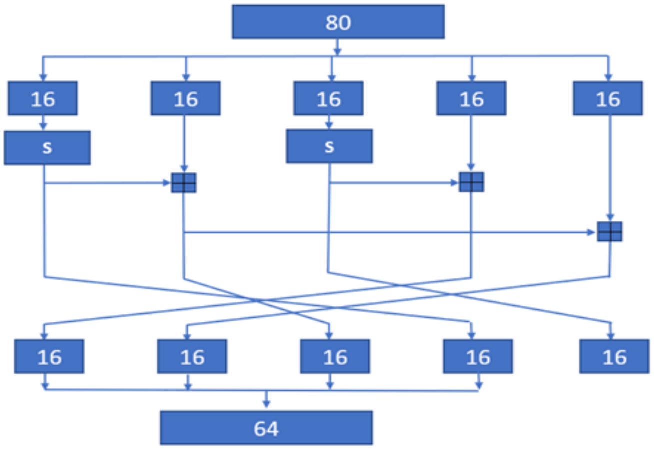


Fig. 4 Algorithm for generating round keys.

IV. STATISTICAL ANALYSIS OF CIPHERTEXTS

One of the main ways to test a block encryption algorithm for security is to conduct statistical analysis since most cryptographic attacks are based on the search for statistical vulnerabilities in the ciphertext.

To test sequences for randomness, there are a large number of algorithms, and for the convenience of checking sequences, software products have already been implemented that contain some sets of tests. Among them, the most common are the tests proposed by NIST, DIEHARD, CRYPT-X, D. Knuth, and others [20].

For statistical analysis of ciphertexts obtained using the ISL-LWC, Present, and Speck encryption algorithms, the NIST and D. Knuth test sets were used.

1) NIST statistical tests: NIST has developed a number of statistical tests which are based on the task of calculating a statistic that characterizes a certain property of a sequence compared with a reference statistic. Reference statistics are obtained mathematically, which is the subject of many theorems and scientific papers on cryptography, probability theory, and number theory. NIST tests have already been used to study the output sequences of cryptographic systems [21]. The tests are based on the concept of the null hypothesis. The null hypothesis is the assumption that there is some relationship between the occurrence of numbers. In other words, the null hypothesis is the assumption that the sequence is truly random (the symbols of which appear equally likely and independently of each other). Therefore, if such a hypothesis is true, then the encryption algorithm will perform well statistically.

To obtain the results of testing ciphers 15 NIST statistical tests were used: frequency bit test, frequency block test, test for a sequence of identical bits, test for the longest sequence of ones in a block, test for binary matrix ranks, spectral test, test for matching non-overlapping patterns, overlapping pattern matching test, Maurer's universal statistical test, linear complexity test, periodicity test, approximate entropy test, cumulative sums test, arbitrary variance test, and another arbitrary variance test [22].

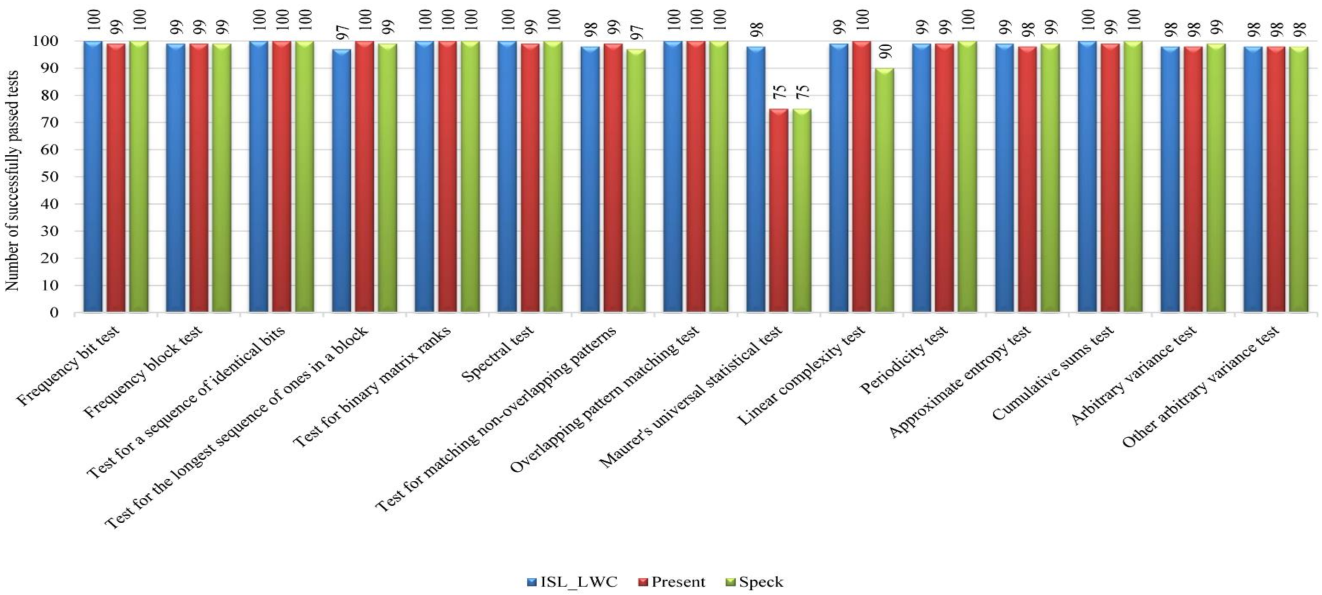


Fig. 5 Comparative analysis of successfully passed NIST tests.

To study the statistical security of the ISL-LWC, Present, and Speck encryption algorithms using NIST tests, each algorithm encrypted 20 files, differing in size, on five different keys. As a result, 100 files were encrypted with each algorithm. The number of successfully passed NIST tests and a comparative analysis of the ISL-LWC, Present, and Speck encryption algorithms are shown in Fig. 5.

In each test, a so-called P-value is calculated, which indicates the level of randomness. If the P-value = 1, then the sequence is perfectly random, and if it is zero, then the sequence is completely predictable. Next, the P-value is compared with the threshold level of randomness α, and if it is greater than α, then the null hypothesis is accepted and the sequence is recognized as random, otherwise, it is recognized as non-random.

In the tests, α = 0.01 is assumed. Therefore:

- If the P-value ≥ 0.01, then the sequence is considered random with a confidence level of 99%;
- If the P-value < 0.01, then the sequence is considered non-random with a confidence level of 99%.

As a result of the study and comparative analysis of the three encryption algorithms according to NIST tests, it was found that the percentage of successfully passed tests by algorithms is: ISL-LWC – 99%, Present – 97.5%, Speck – 97%. From the obtained results, we can conclude that the ISL-LWC algorithm satisfies the statistical security criteria.

2) Statistical tests by D. Knuth: One of the first sets of statistical tests was proposed by D. Knuth in 1969 and described in his classic work "The Art of Computer Programming". D. Knuth's set contains such tests as the serial test, gap test, poker test, coupon collector test, permutation test, monotonicity test, and correlation test. The tests are based on the chi-square (χ^2) statistical test. The calculated value of the χ^2 statistic is compared with the tabular results and, depending on the probability of occurrence of such a statistic, a conclusion is made about its quality [23]. Among the advantages of these tests are their small number and the existence of fast execution algorithms. The disadvantage is the uncertainty in the interpretation of the results [24].

To study the statistical security of the ISL-LWC, Present, and Speck encryption algorithms using the D. Knuth tests, we encrypted with each algorithm the same 100 files that were checked using the NIST tests. The number of successfully passed the tests by D. Knuth and a comparative analysis of the ISL-LWC, Present, and Speck encryption algorithms are shown in Fig. 6.

As a result of the study on the tests of D. Knuth and a comparative analysis of the three encryption algorithms, it was found that the percentage of successfully passed tests by the algorithms is 93.5% for ISL-LWC, 99% for Present, and 99% for Speck. From the obtained results, we can conclude that the ISL-LWC algorithm satisfies the statistical security criteria.

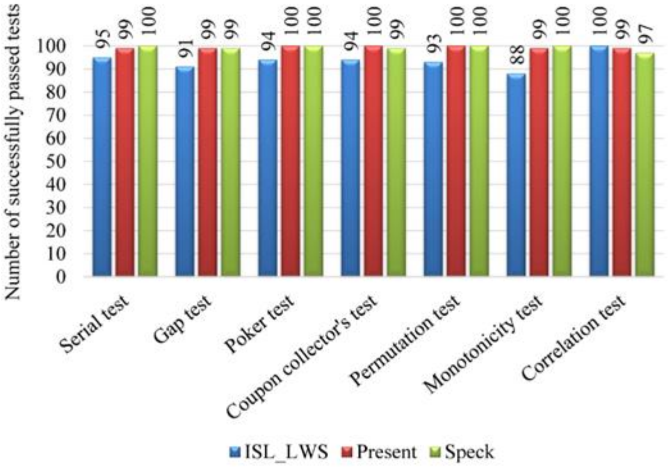


Fig. 6 Comparative analysis of successfully passed tests by D. Knuth.

3) *Study of statistical security indicators:* For the study of statistical security, the following indicators were considered:

- average number of output bits that change when one input bit changes (avalanche effect);
- degree of completeness (d_c);
- degree of avalanche effect (d_a);
- degree of strict avalanche criterion (d_{sa}).

They are considered for various numbers of cycles and randomly taken encryption keys. The definition of the above indicators is presented in [25].

The essence of the experiment is to evaluate the depth of the avalanche effect of the ISL-LWC encryption algorithm, which is determined by the number of encryption rounds. The experiment was carried out on 100, 1000, and 10,000 blocks of ciphertext obtained using the ISL-LWC algorithm. Table IV presents the results of the assessment of the statistical security indicators of the ISL-LWC cipher.

TABLE IV RESULTS OF THE ASSESSMENT OF STATISTICAL SECURITY INDICATORS OF THE ISL-LWC CIPHER

Round number	D_{min}	D_{max}	d_w	M_{min}	M_{max}	m_w	d_c	d_a	d_{sa}
ISL-LWC (100 blocks)									
1	9.0020	10.8636	9.9328	3.07485	5.7076	4.3912	0.125	0.1372	0.0867
2	50.6934	52.5550	51.6242	11.30266	13.9354	12.61906	0.4682	0.3943	0.3314
3	64.6894	66.5510	65.6202	20.8173	23.4501	22.1337	0.8593	0.6916	0.6345
4	43.9609	45.8226	44.8918	27.3195	29.9523	28.6359	0.9843	0.8939	0.8350
5	20.9659	22.8276	21.8968	30.1476	32.7804	31.4640	1	0.9776	0.9114
6	15.5450	17.4067	16.4759	30.5796	33.2124	31.8960	1	0.9898	0.9194
7	14.8608	16.7225	15.7916	30.7411	33.3739	32.0575	1	0.9887	0.9219
8	14.4042	16.2658	15.3350	30.6254	33.2582	31.9418	1	0.9894	0.9203
9	14.7902	16.6519	15.7211	30.5853	33.2181	31.9017	1	0.9905	0.9176
10	15.1740	17.0357	16.1049	30.6101	33.2429	31.9265	1	0.9902	0.9218
11	15.4715	17.3332	16.4024	30.6745	33.3073	31.9909	1	0.9911	0.9196
12	15.1822	17.0439	16.1131	30.7321	33.3649	32.0485	1	0.9926	0.9216
13	14.9554	16.8170	15.8862	30.6618	33.2946	31.9782	1	0.9903	0.9211
14	15.0505	16.9122	15.9813	30.5975	33.2003	31.8839	1	0.989086	0.9199
15	14.6191	16.4812	15.5504	30.6261	33.2589	31.9425	1	0.9898	0.9224
16	14.8984	16.7601	15.8292	30.6467	33.2795	31.9631	1	0.9893	0.9201
ISL-LWC (1000 blocks)									
1	9,0020	10,8637	9,9329	3,0749	5,7077	4,3913	0,1250	0,1372	0,0867
2	50,6934	52,5551	51,6243	11,3027	13,9355	12,6191	0,4683	0,3943	0,3314
3	64,6894	66,5511	65,6202	20,8174	23,4502	22,1338	0,8594	0,6917	0,6345
4	43,9610	45,8227	44,8918	27,3195	29,9523	28,6359	0,9844	0,8940	0,8351
5	20,9660	22,8277	21,8968	30,1477	32,7805	31,4641	1	0,9777	0,9115
6	15,5451	17,4068	16,4759	30,5797	33,2125	31,8961	1	0,9899	0,9195
7	14,8609	16,7225	15,7917	30,7411	33,3739	32,0575	1	0,9888	0,9219
8	14,4042	16,2659	15,3351	30,6255	33,2583	31,9419	1	0,9895	0,9204
9	14,7903	16,6520	15,7211	30,5853	33,2181	31,9017	1	0,9906	0,9177
10	15,1741	17,0358	16,1049	30,6102	33,2430	31,9266	1	0,9902	0,9218
11	15,4716	17,3333	16,4024	30,6745	33,3073	31,9909	1	0,9911	0,9197
12	15,1823	17,0439	16,1131	30,7322	33,3650	32,0486	1	0,9927	0,9217
13	14,9554	16,8171	15,8862	30,6619	33,2947	31,9783	1	0,9904	0,9212
14	15,0505	16,9122	15,9814	30,5975	33,2003	31,8839	1	0,9891	0,9200
15	14,6191	16,4813	15,5504	30,6261	33,2589	31,9425	1	0,9899	0,9225
16	14,8984	16,7601	15,8293	30,6467	33,2795	31,9631	1	0,9894	0,9202
ISL-LWC (10000 blocks)									
1	9,0020	10,8637	9,9329	3,0749	5,7077	4,3913	0,1250	0,1372	0,0867
2	50,6934	52,5551	51,6243	11,3027	13,9355	12,6191	0,4683	0,3943	0,3314
3	64,6894	66,5511	65,6202	20,8174	23,4502	22,1338	0,8594	0,6917	0,6345
4	43,9610	45,8227	44,8918	27,3195	29,9523	28,6359	0,9844	0,8940	0,8351
5	20,9660	22,8277	21,8968	30,1477	32,7805	31,4641	1	0,9777	0,9115
6	15,5451	17,4068	16,4759	30,5797	33,2125	31,8961	1	0,9899	0,9195

Round number	D_{min}	D_{max}	d_w	M_{min}	M_{max}	m_w		d_c		d_a		d_{sa}	
7					14,8609	16,7225	15,7917	30,7411	33,3739	32,0575	1	0,9888	0,9219
8					14,4042	16,2659	15,3351	30,6255	33,2583	31,9419	1	0,9895	0,9204
9					14,7903	16,6520	15,7211	30,5853	33,2181	31,9017	1	0,9906	0,9177
10					15,1741	17,0358	16,1049	30,6102	33,2430	31,9266	1	0,9902	0,9218
11					15,4716	17,3333	16,4024	30,6745	33,3073	31,9909	1	0,9911	0,9197
12					15,1823	17,0439	16,1131	30,7322	33,3650	32,0486	1	0,9927	0,9217
13					14,9554	16,8171	15,8862	30,6619	33,2947	31,9783	1	0,9904	0,9212
14					15,0505	16,9122	15,9814	30,5975	33,2003	31,8839	1	0,9891	0,9200
15					14,6191	16,4813	15,5504	30,6261	33,2589	31,9425	1	0,9899	0,9225
16					14,8984	16,7601	15,8293	30,6467	33,2795	31,9631	1	0,9894	0,9202

In Table IV, the following designations are used:

- M_{min} is the minimum value of the mathematical expectation of the number of changed bits for some bit at the input;
- M_{max} is the maximum value of the mathematical expectation of the number of changed bits for some bit at the input;
- D_{min} and D_{max} are the variances of the number of changed bits in the bitwise estimation of the minima and maxima of the mean values;

m_w is the average number of changed bits:

$$m_w = \frac{M_{min}+M_{max}}{2} \tag{1}$$

Analyzing the data obtained in Table IV, we can conclude that with an increase in the number of blocks for encryption, more accurate values of d_a and d_{sa}, are obtained, i.e. they approach value 1 faster in the fourth and subsequent rounds. As a result of the study, it was found that at the 4th round of encryption of the ISL-LWC algorithm, the input sequence is completely confused.

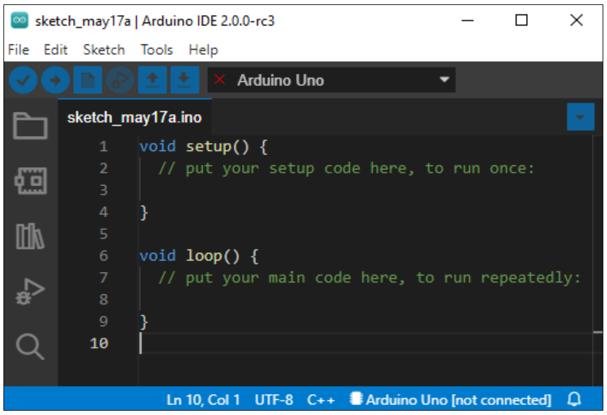
Results of the study and comparative analysis of the time of encryption and key generation on the Arduino Uno R3 board.

Encryption time testing for three encryption algorithms Speck, Present, and ISL-LWC was carried out on the Arduino Uno R3 board (Fig. 7).

- main features of Arduino Uno R3;
- microcontroller - ATmega328;
- clock frequency - 16 MHz;
- operating voltage - 5 V;
- flash memory - 32 MB;
- RAM - 2 Kb.

Fig. 7 Arduino Uno R3 board.

In [25] Arduino IDE version 2.0.0-rc3 was used to compile and upload the source code of lightweight encryption algorithms to the Arduino Uno R3 board (Fig. 8).



sketch_may17a | Arduino IDE 2.0.0-rc3
File Edit Sketch Tools Help
Arduino Uno
sketch_may17a.ino
Ln 10, Col 1 UTF-8 C++ Arduino Uno [not connected]

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}

```

Fig. 8 Arduino IDE.

The three encryption algorithms (Speck, Present, and ISL-LWC) were implemented by the staff of the Information Security Laboratory of the Institute of Information and Computational Technologies of the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (RK MSHE CS IICT ISL) in the high-level programming language C++.

The tests were carried out on open-source software platforms developed by the US National Institute of Standards and Technology in order to unify, simplify, and speed up the testing of lightweight cryptographic algorithms.

The results of the study and comparative analysis of the Present, Speck, and ISL-LWC algorithms are shown in Table V.

TABLE V COMPARATIVE ANALYSIS OF THE ALGORITHMS BY THE TIME OF ENCRYPTION AND KEY GENERATION

Encryption algorithm	Key size, bits	Plaintext block size, bits	Encryption time, μs	Key setting time, μs
Present	80	64	2111.56	1541.31
Speck	96		16.90	1320.69
ISL-LWC	80		108.59	275.12

As a result of a comparative analysis of Table V, it was found that the proposed encryption algorithm works faster than Present, and when scheduling round keys, it is 6 and 5 times faster than the algorithms under consideration, respectively.

V. Conclusion

Lightweight encryption algorithms are considered a relatively new direction in the development of symmetric cryptography. This need arose as a result of the emergence of a large number of devices with little computing power and memory. Therefore, there was a need to develop algorithms that can provide a sufficient level of security with minimal use of resources.

This paper provides a brief literature review of existing lightweight encryption algorithms. A new lightweight block encryption algorithm ISL-LWC, developed by the staff of the RK MSHE CS IICT ISL, is presented.

The cryptographic properties of the developed algorithm were studied using the evaluation of the "avalanche effect" and statistical tests. Based on the work carried out, it was found that the proposed encryption algorithm is effective in providing a good avalanche effect, and the encrypted data is close to random and is statistically safe.

The developed algorithm is implemented in software and hardware on the Arduino Uno R3 board. A study and comparative analysis of the encryption and key generation time with the well-known lightweight algorithms Present and Speck have been carried out.

The obtained test results allow us to conclude that the ISL-LWC cipher is generally not inferior to these two well-known lightweight algorithms. Further study of the cryptographic properties of this algorithm by other methods, such as linear and differential cryptanalysis, etc., will be continued. The results will be presented in subsequent papers and used to improve the proposed algorithm.

Acknowledgment

The work was performed within the framework of the grant funding project AP09259570 “Development and study of a domestic lightweight encryption algorithm with limited resources” of the RK MSHE CS.

References

[1] V. A. Dovgal, and D. V. Dovgal, “Internet of Things: Concept, Applications, and Tasks,” Bulletin of the Adyghe State University, Series 4: Natural-Mathematical and Technical Sciences, vol. 1, no. 212, pp. 129-135, 2018.

[2] F. Chetouane, “An Overview on RFID Technology Instruction and Application,” IFAC-PapersOnLine, vol. 48, no. 3, pp. 382-387, 2015, https://doi.org/10.1016/j.ifacol.2015.06.111.

[3] H. Hasan, G. Ali, W. Elmedany, and C. Balakrishna, “Lightweight Encryption Algorithms for Internet of Things: A Review on Security and Performance Aspects,” Int. Con. on Innov. and Intel. for Inf, Com. and Tech (3ICT), pp. 239-244, 2022, doi: 10.1109/3ICT56508.2022.9990859.

[4] P. K. Dhillon, and S. Kalra, “A lightweight biometrics based remote user authentication scheme for IoT services,” Jour. Inf. Sec. and Appl, vol. 34, pp. 255–270.

[5] T. Eisenbarth, and S. Kumar, “A Survey of Lightweight-Cryptography Implementations,” IEEE Des Test Com., vol. 24, no. 6, pp. 522–533, 2007.

[6] I. Bhardwaj, A. Kumar, and M. Bansal, “A review on lightweight cryptography algorithms for data security and authentication in IoTs, “ 2017 4th Int. Conf. on Sig. Proc., Com. and Cont., (ISPCC), India, pp. 504-509, 2017, doi: 10.1109/ISPCC.2017.8269731.

[7] A. E. Zhukov, “Lightweight Cryptography [Part 1. Cybersecurity Issues],” vol. 1, no. 9, pp. 26-43, 2015.A. S. Soskov, and B. Ya. Ryabko, “The distinguishing attack on ARX-based lightweight block ciphers,” Comp. Tech. vol. 24, no. 3, pp. 106–116, 2019. DOI: 10.25743/ICT.2019.24.3.008.

[8] E. A. Ischukova, and E. A. Tolomanenko, “Analysis of the algorithms for encryption of lightweight cryptography in the context of the Internet of Things,” Mod. High Tech., vol. 3, no. 2, pp. 182-186, 2019, URL: https://top-technologies.ru/ru/article/view?id=37462.

[9] Zh. Tang, J. Cui, H. Zhong, and M. Yu, “A Random PRESENT Encryption Algorithm Based on Dynamic S-box International,” Jour. of Sec. and Its Appl., vol. 10, no. 3, pp. 383-392, 2016, http://dx.doi.org/10.14257/ijsia.2016.10.3.33.

[10] A. Suhail, N. Mir, A. Mehvish, S. Ishfaq, and B. M. Tariq, “FPGA Implementation of PRESENT Block Cypher with Optimised Substitution Box,” 2022 Smart Tech., Com. and Robot. STCR, pp. 1-6, 2022, doi: 10.1109/STCR55312.2022.10009366.

[11] T. Shirai, T. Shibutani, and K. Akishita, “The 128-bit block cipher CLEFIA,” FSE 2007. LNCS, vol. 4593, pp. 181–195, 2007.

[12] F. M. Qatan, and I. W. Damaj, “High-speed KATAN ciphers on-a-chip,” Comp. sys. and Ind. Inf. ICCSII, 2012 Inter. Conf, IEEE, pp. 1–6, 2012.

[13] E. Aysu, and P. Gulcan, “Schaumont. SIMON says: Break area records of block ciphers on FPGAs,” IEEE Emb. Syst Lett, vol. 6, pp. 37–40, 2014, https://doi.org/10.1109/les.2014.2314961.

[14] R. Beaulieu, S. D. Treatman-Clark, B. Shors, J. Weeks, Smith, and L. Wingers, “The SIMON and SPECK lightweight block ciphers,” 2015 52nd ACM/EDAC/IEEE Des. Auto. Conf DAC, San Francisco, USA, 2015, pp. 1-6, doi: 10.1145/2744769.2747946.

[15] U. Muhammad, A. Irfan, M. Imran, Kh. Shujaat, and A. Sh. Usman, “SIT. A Lightweight Encryption Algorithm for Secure Internet of Things,” IJACSA Inter. Jour. of Adv. Comp. Sci. and Appl, vol. 8, no. 1, pp. 402-411, 2017.

[16] F. Xinxin, M. Kalikinkar, and G. Guang, “WG-8: A Lightweight Stream Cipher for Resource-Constrained Smart Devices Quality, Reliability, Security and Robustness in Heterogeneous Networks,” vol. 115, pp. 617–632, 2013, https://doi.org/10.1007/978-3-642-37949-9_54.

[17] A. Bogdanov, L. Knudsen, G. Leander, and et al, “PRESENT: An ultra-lightweight block cipher,” CHES 2007. LNCS, vol. 4727, pp. 450–466, 2007.

[18] A. Khompysh, N. Kapalova, K. Algazy, D. Dyusenbayev, and K. Sakan, “Design of substitution nodes (S-Boxes) of a block cipher intended for preliminary encryption of confidential information,” Cogent Engineering, vol. 9, no. 1, pp. 1-14, 2022, DOI: 10.1080/23311916.2022.2080623.

[19] A. A. Perov, “Using NIST statistical tests for the analysis of the output sequences of block ciphers,” Sci. Bull. of NSTU, vol. 3, no. 76, pp. 87–96, 2019. doi:10.17212/1814-1196-2019-3-87-96.

[20] F. Sulak, M. Uğuz, O. Koçak, and A. Doğanaksoy, “On the independence of statistical randomness tests included in the NIST test suite,” Turk. Jour. of Elec. Engin. & Comp. Scien. vol. 25, no.5, pp. 3673-3683, 2017. doi:10.3906/elk-1605-212.

[21] M. O. Pikuza, and S. Yu. Mikhnevich, “Testing a hardware random number generator using NIST statistical test suite,” BSUIR Reports, vol. 19, no. 4, pp. 37-42, 2021. https://doi.org/10.35596/1729-7648-2021-19-4-37-42.

[22] K. Sakan, S. Nyssanbayeva, N. Kapalova, K. Algazy, A. Khompysh, and D. Dyusenbayev, “Development and analysis of the new hashing algorithm based on a block cipher,” Easter-Euro. Jour. of Enter. Techn, vol. 2, no. 9 (116), pp. 60–73, 2022. https://doi.org/10.15587/1729-4061.2022.252060 2022

[23] N. A. Kapalova, A. Khompysh, A. Müslüm, and K. Algazy, “A block encryption algorithm based on exponentiation transform [Cogent

Engineering], 2020, Vol.7, no. 1, pp.1-12, https://doi.org/10.1080/23311916.2020.1788292

[24] I. V. Lisitskaya, A. A. Nastenko, K. E. Lissitzky, “ Large ciphers - random substitutions. Comparison of statistical security indicators of block symmetric ciphers submitted to the Ukrainian competition,” East. Euro. Jour. of Adv. Tech. ISSN 1729-3774 vol. 6, no.9 (60), pp. 1-11, 2012.

[25] M. Simon, “Programming Arduino: Getting Started with Sketches,” Third Edition, McGraw Hill LLC, p.176, 2022.