INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2021, VOL. 67, NO. 1, PP. 127-132
Manuscript received August 17, 2020; revised January, 2021. DOI: 10.24425/ijet.2021.135954

Development and Analysis of Symmetric Encryption Algorithm Qamal Based on a Substitution-permutation Network

Rustem G. Biyashev, Nursulu A. Kapalova, Dilmuhanbet S. Dyusenbayev, Kunbolat T. Algazy, Waldemar Wojcik, and Andrzej Smolarz

***Abstract*—This paper represents a developed cryptographic information protection algorithm based on a substitution-permutation network. We describe the cryptographic transformations used in the developed algorithm. One of the features of the algorithm is the simplicity of its modification with regard to different security levels. The algorithm uses a pre-developed S-box tested against differential and linear cryptanalysis. The S-box is consistent with one of the known standards AES and GOST R 34.12-2015. We provide the findings of an avalanche-effect investigation and statistical properties of ciphertexts. The algorithm actually meets the avalanche-effect criterion even after the first round.**

***Keywords*—encryption, cryptanalysis, S-box, SP-network, avalanche effect, statistical property**

I. Introduction

SYMMETRIC block encryption algorithms are today the principal cryptographic tool for ensuring confidentiality in data processing in up-to-date information and telecommunication systems [1,2]. Modern symmetric block ciphers are mainly built based on two approaches: Feistel network and substitution-permutation network (SP-network). As is known, ciphers are based on reversible transformations of plaintext. When working on ciphers, care must be exercised that every operation performed is both cryptographically robust and reversible under a known key [3].

Present-day ciphers are based on the Kerckhoffs' Principle [4] that the security of a cipher is ensured by keeping secret the key, but not the encryption algorithm. From the viewpoint of an adversary, a secure cryptosystem is a "black box", the input and output information sequences of which are mutually independent, provided that the output encrypted sequence is pseudorandom [5,6]. Thus, a ciphertext obtained is investigated for pseudorandomness by using statistical tests (testing). It is analyzed the dependence of changes in the ciphertext when changing characters or bits in the original plaintext or key. Different types of such an analysis are aimed to detect statistical particularities or any dependence between characters of the plaintext and the ciphertext.

In the cryptographic information protection facilities in use, the length of a message secured with a symmetric block cipher generally far exceeds the length of an encryption key. In this situation, the criterion of unconditional security of the utilized cipher is not fulfilled [5-9]. Against this background, the strength of an encryption algorithm is based on the assumption that an adversary has time and computer power limits. This implies the definition of practical strength criterion, i.e. it is impossible for a long time to implement an attack on a cipher within the conditions of present-day computing base.

Block ciphers are also used as a base unit to build other cryptographic algorithms (primitives), such as pseudorandom sequence generators (PRNG), stream ciphers, and hash functions. The level of strength and the properties of the symmetric block encryption algorithm in use govern to a large extent the strength of cryptographic information protection facilities, the security of cryptographic protocols, and protection of an information and communication system as a whole [5-9].

A secure block cipher should meet certain conditions. These conditions were given by Claude Shannon in a number of his fundamental papers on the theory of encryption [10-12]. A secure cipher should have the properties of diffusion and confusion.

Diffusion means that one character (bit) of an input plaintext affects several characters (bits) of the resulting ciphertext, ideally, all the characters within one block. If this condition is fulfilled, then the encryption of two data blocks with minor differences between them should produce two completely different blocks of ciphertext. The same requirement should be held between ciphertext and key, i.e. one character (bit) of the key should affect several characters (bits) of the ciphertext. Diffusion obscures relationships between the ciphertext and the original text.

Confusion refers to the property of a cipher to obscure the connections between characters of the original text and its ciphertext. If a cipher produces a reasonably good "confusion" of the bits of the original text, then the respective ciphertext does not feature any statistical or functional regularity. Confusion obscures the relationship between the encrypted text and the key.

This work was supported within the framework of the project BR05236757 “Development of software and software-hardware means for cryptographic protection of information during its transmission and storage in general-purpose info-communication systems and networks”, which is being implemented at the Institute of Information and Computer Technologies.

Rustem. G. Biyashev (e-mail: brg@ipic.kz), Nursulu A. Kapalova (e-mail: nkapalova@mail.ru), Dilmuhanbet S. Dyusenbayev (e-mail: dimash_dds@mail.ru), Kunbolat T. Algazy (e-mail: kunbolat@mail.ru) are with with Institute of Information and Computational Technologies of the Committee of Science of the Ministry of Education and Science of the Republic of Kazakhstan, Almaty;

Waldemar Wojcik (e-mail: waldemar.wojcik@pollub.pl), Andrzej Smolarz (e-mail: a.smolarz@pollub.pl) are with Lublin University of Technology, Lublin, Poland.

In view of the above, a symmetric block encryption algorithm based on an SP-network was developed. We called the new algorithm Qamal.

II. Encryption Algorithm Qamal

The block diagram of the developed encryption algorithm is presented in Figure 1. The algorithm supports the sizes of block and key of 128, 192 and 256 bits. The number of encryption rounds depends on the size of the block and key. For keys K with the length of 128, 192, and 256 bits, the number of encryption rounds is 8, 10, and 12 respectively. All rounds are completed with modulo 2 additions to the round key. The encryption algorithm includes the developed procedures (primitives) of key applying by bitwise addition (XOR), substitution S-box, and mixing procedures Mixer1 and Mixer2.

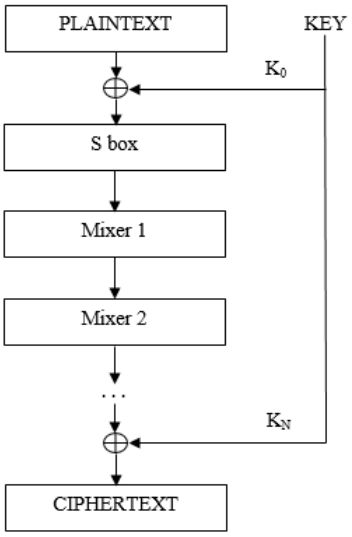

Figure 1. Qamal encryption algorithm block diagram

In the first procedure, the operation of applying (addition to) a key modulo 2 (XOR operation) on a plaintext block is performed.

The second procedure is the formation of a substitution S1-box, where a nonlinear transformation on bytes is performed, i.e. a nonlinear bijective substitution is applied to each byte. The resulting S1-box is shown in Table I.

The third procedure is the formation of Mixer1 box. The box bytes are represented by a two-dimensional array A of size $m \times 4$, where m takes the value of 4, 6 or 8 depending on the initial block size.

$$A = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ . & . & . & . \\ a_{m0} & a_{m1} & a_{m2} & a_{m3} \end{bmatrix}.$$

The bytes of each column are added to each other modulo 256:

$$M_1(b_{ij}) = \sum_{i=0}^{m} a_{ij} \bmod 256, j = \overline{0,3}.$$

Then the new byte obtained in the first column replaces the uppermost byte a_{00}, while all the original bytes of the column rotate downshift of one position. This operation is repeated m times. As a result, we get m new bytes in the first column. Next, the operation is performed for the other three columns (Figure 2).

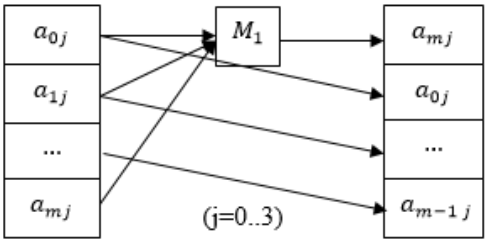

Figure 2. Mixer1 box workflow

Table I.
Substitution table for S1-box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	C9	34	F0	18	55	86	21	6B	87	D2	6E	99	BD	31	98	89
1	29	73	83	8B	1A	19	E1	E4	F3	5B	72	3F	A6	F9	2E	A3
2	7E	10	94	07	EC	AD	2F	26	20	93	66	3D	DD	64	5F	C1
3	13	E0	80	25	D3	08	75	6A	B9	2D	D1	CC	FD	CA	3B	FC
4	D5	DA	E2	CE	A0	7F	AE	C8	9C	09	3C	95	BA	35	3E	7B
5	FA	8D	23	AB	D9	E8	74	2A	C3	A8	D8	52	45	B5	0A	0C
6	A4	61	9A	FB	AA	F6	78	84	C4	E9	EE	54	50	81	DF	90
7	36	B4	BB	44	C5	96	4B	28	14	E6	8F	FF	B0	1F	53	47
8	00	4C	40	2C	9B	9F	4A	01	7D	AF	92	56	7A	DB	8E	16
9	63	24	A9	1D	33	4D	E7	1C	70	69	B7	C6	32	E5	57	03
A	97	A5	EB	D4	BC	5D	F8	85	06	F2	59	F4	17	22	38	DC
B	0B	FE	BE	CD	41	82	04	0E	48	71	30	AC	EF	C7	2B	CB
C	B8	8C	5A	42	A7	4E	D0	46	BF	B3	91	E3	11	7C	6F	DE
D	88	58	1E	5C	9D	60	C0	62	05	79	ED	76	C2	02	65	D7
E	F1	8A	77	F7	37	B1	0F	67	CF	0D	A1	6C	4F	3A	39	1B
F	27	B6	5E	F5	EA	6D	15	9E	B2	12	A2	68	43	51	49	D6

The fourth procedure is the transformation of Mixer2. As a consequence of the formation of Mixer1 box, we get the new array B of size $m \times 4$, where m takes on values of 4, 6 or 8 depending on the block size:

$$B = \begin{bmatrix} b_{00} & b_{01} & b_{02} & b_{03} \\ b_{10} & b_{11} & b_{12} & b_{13} \\ . & . & . & . \\ b_{m0} & b_{m1} & b_{m2} & b_{m3} \end{bmatrix}.$$

Each row of the array is represented in the form of a cubic polynomial with the coefficients from the finite field $GF(2^8)$. These polynomials appear as follows:

$$b_i(x) = b_{i0}x^3 + b_{i1}x^2 + b_{i2}x + b_{i3}, i = 0,\ldots,3.$$

Each polynomial $b_i(x)$ multiplies by fixed (preselected) polynomials $m_i(x)$ modulo $p(x)$:

$$m_0(x) = 168\ x^3 + 34x^2 + 187x + 186,$$
$$m_1(x) = 210x^3 + 53x^2 + 210x + 101,$$
$$m_2(x) = 218x^3 + 25x^2 + 150x + 210,$$
$$m_3(x) = 144x^3 + 75x^2 + 158x + 27,$$
$$m_4(x) = 163x^3 + 4x^2 + 111x + 106,$$
$$m_5(x) = 150x^3 + 237x^2 + 13x + 53,$$
$$m_6(x) = 99x^3 + 59x^2 + 104x + 205,$$
$$m_7(x) = 167x^3 + 49x^2 + 241x + 154,$$
$$p(x) = x^4 + x + 55.$$

The polynomials $m_i(x)$ are used in the following manner. If the size of the plaintext block is 128 bits, then the first four polynomials $m_0(x)$, $m_1(x)$, $m_2(x)$, $m_3(x)$ are selected. For the block size of 192 bits, the first 6 polynomials $m_0(x)$,

$m_1(x)$, $m_2(x)$, $m_3(x), m_4(x), m_5(x)$ are taken. For the third possible block size, all the eight polynomials are used.

III. Decryption Algorithm Qamal

To decrypt a ciphertext, all the cryptographic transformations used for encryption are inverted and applied in the decryption algorithm in reverse order. Round keys are also used in reverse order. In the process of decryption, for each above-mentioned block size, it is performed respectively 8, 10 and 12 rounds with inverse operations InvS, InvMixer1, and InvMixer2 in each round.

Operation InvS is the inverse of the operation of obtaining elements in the S-box. Bytes of the S-box array are replaced with new bytes obtained through the inverse substitution. As a result, we get the inverse S-box.

Operation InvMixer1 is inverse of the transformation $M_1(b_{ij})$.

Operation InvMixer2 is a procedure inverse to the one for obtaining Mixer2 box. To obtain the inverse box of Mixer2, each row of the array is considered as a four-termed polynomial over $GF(2^8)$. This polynomial multiplies by fixed polynomials modulo polynomial $p(x)$:

$$\mathrm{m}_0^{-1}(\mathrm{x}), \mathrm{m}_1^{-1}(\mathrm{x}), \mathrm{m}_2^{-1}(\mathrm{x}), \mathrm{m}_3^{-1}(\mathrm{x}),$$
$$\mathrm{m}_4^{-1}(\mathrm{x}), \mathrm{m}_5^{-1}(\mathrm{x}), \mathrm{m}_6^{-1}(\mathrm{x}), \mathrm{m}_7^{-1}(\mathrm{x}).$$

Table II
InvS (S1-box inversion)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	80	87	DD	9F	B6	D8	A8	23	35	49	5E	B0	5F	E9	B7	E6
1	21	CC	F9	30	78	F6	8F	AC	03	15	14	EF	97	93	D2	7D
2	28	06	AD	52	91	33	27	F0	77	10	57	BE	83	39	1E	26
3	BA	0D	9C	94	01	4D	70	E4	AE	EE	ED	3E	4A	2B	4E	1B
4	82	B4	C3	FC	73	5C	C7	7F	B8	FE	86	76	81	95	C5	EC
5	6C	FD	5B	7E	6B	04	8B	9E	D1	AA	C2	19	D3	A5	F2	2E
6	D5	61	D7	90	2D	DE	2A	E7	FB	99	37	07	EB	F5	0A	CE
7	98	B9	1A	11	56	36	DB	E2	66	D9	8C	4F	CD	88	20	45
8	32	6D	B5	12	67	A7	05	08	D0	0F	E1	13	C1	51	8E	7A
9	6F	CA	8A	29	22	4B	75	A0	0E	0B	62	84	48	D4	F7	85
A	44	EA	FA	1F	60	A1	1C	C4	59	92	64	53	BB	25	46	89
B	7C	E5	F8	C9	71	5D	F1	9A	C0	38	4C	72	A4	0C	B2	C8
C	D6	2F	DC	58	68	74	9B	BD	47	00	3D	BF	3B	B3	43	E8
D	C6	3A	09	34	A3	40	FF	DF	5A	54	41	8D	AF	2C	CF	6E
E	31	16	42	CB	17	9D	79	96	55	69	F4	A2	24	DA	6A	BC
F	02	E0	A9	18	AB	F3	65	E3	A6	1D	50	63	3F	3C	B1	7B

IV. Round Key Generation Algorithm

Round keys K_i are generated from the cipher key K with the use of the key extension procedure. Eventually, we get an array of round keys, which is then used to select a needed round key. The scheme for obtaining round keys is presented in Figure 3.

The procedure of generating round keys involves all the transformations used in the process of encryption, save a different substitution table (S2-box, Table III), and a new transformation *Module $p_i(x)$*.

Module $p_i(x)$. Let $p_1(x)$, $p_2(x), \ldots, p_S(x)$ be binary irreducible polynomials used as working bases (not to be confused with modulo $p(x)$ used in Mixer2), and $P(x) = p_1(x)\, p_2(x) \ldots p_S(x)$. The polynomial $P(x)$ degree $N = m_1 + m_2 + \cdots + m_S$ is equal to the block size (i.e. 128, 192, 256). The output data from the Mixer2 box we represent in the form of binary polynomial $N(x)$. Here $k_1(x), k_2(x), \ldots, k_s(x)$ are remainders of dividing polynomial $N(x)$ by respective bases $p_i(x), i = 1, \ldots, s$, where $p_i(x), i = 1, \ldots, s$ are secret elements of the key schedule procedure.

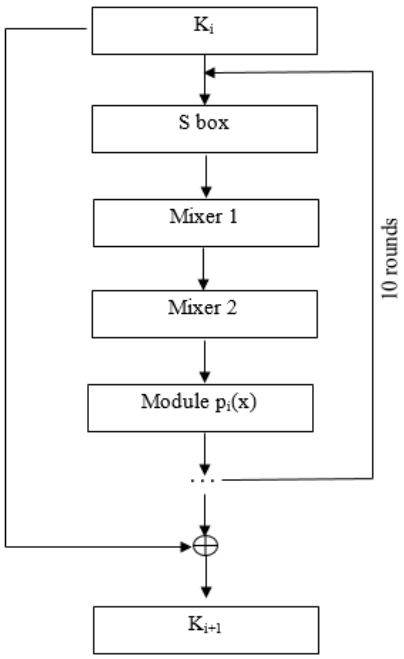

Figure 3 – Key K_i extension scheme, where i =0,1,…,6 (8,10)

V. Experimental Tests of the Avalanche Effect

When developing encryption algorithms, it is imperative to analyze them for their strength against various types of cryptographic attacks. Among the most common standard methods at present are attacks based on linear and differential cryptanalysis. The essence of the latter is to track the change in the difference between the values of the output bits depending on the change in the input bits (in the original data) on different rounds of transformation. A necessary condition for ensuring the strength of an encryption algorithm against differential cryptanalysis is the presence of the avalanche effect in the basic transformation.

The avalanche effect is an important cryptographic property for encryption, wherein small changes in the input bits or key bits result in avalanche changes in the output ciphertext bits. If an algorithm fails to provide the avalanche effect to a required level, then a cryptanalyst can make predictions about the input data, being given the output. To estimate the degree of the avalanche effect in the transformation, an avalanche parameter was determined and used – the numerical value of the deviation of the probability of a bit change in the output sequence in response to a bit change in the input sequence from the required probability value equal to 0.5 [13]. For the avalanche-effect criterion, the value of the avalanche parameter is determined by the formula $\varepsilon = |2\mathrm{k_i} - 1|$, where i is the sequence number of the altered bit in the input, k_i is the probability that half the output bits are changed following a change in the i-th input bit compared to the output value for the initial (unchanged) input value.

The formula shows that the extremity ε can take values from 0 to 1 inclusive. The closer ε is to zero, the better is the algorithm. And vice versa, the closer the value of ε is to 1, the weaker is the algorithm.

TABLE III
S2-BOX USED IN GENERATING ROUND KEYS

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	55	A8	78	9C	C3	ED	B1	DE	CD	2C	09	51	27	2D	43	C2
1	CA	45	3A	CE	7B	79	84	7D	BF	E6	69	1F	5E	CB	9E	E2
2	49	38	8E	7C	31	DF	98	42	91	57	90	A6	BD	F1	41	AC
3	20	96	8C	C7	4B	BE	70	E9	D0	4D	1A	A1	B0	DA	5D	D3
4	88	B5	30	47	6B	35	12	B2	B4	17	10	A2	60	9B	0D	FD
5	E4	C6	54	EB	B7	B9	7F	AF	21	5C	D4	99	5F	3E	A9	F3
6	3C	C0	67	13	6A	2F	1C	29	89	58	73	EC	14	39	D8	4E
7	44	02	59	23	F2	0C	FC	AB	74	87	92	36	82	04	16	0E
8	BB	01	F6	15	E7	DC	8F	07	4A	FF	65	1B	25	8B	75	D7
9	A5	7A	A7	FA	24	E5	AE	61	CF	9D	32	66	AA	05	D2	62
A	8D	C4	4F	26	06	0A	D9	7E	F7	E3	F0	34	40	0F	FB	1E
B	6F	A3	D1	BA	95	3D	33	71	83	18	E0	CC	2B	A0	D5	28
C	E1	64	9F	97	4C	A4	76	B3	19	08	68	C1	22	1D	B8	8A
D	E8	50	00	C9	46	56	5A	72	F5	3B	63	94	93	9A	0B	AD
E	DD	C8	FE	5B	53	85	6E	EE	86	80	F9	52	81	11	2A	48
F	C5	EA	EF	DB	B6	3F	37	77	6D	03	2E	D6	F4	BC	F8	6C

The structure of the Qamal encryption algorithm consists of the procedures of key addition using bitwise addition (*XOR*), substitution S-box, and mixing Mixer1 and Mixer2. Consider an example of how the transformations used in the algorithm affect the avalanche effect.

As an input, we take two plaintexts that differ from each other by only one bit. To encrypt them, we use the same key. We find out how this change is diffused to half the block in one round:

Plaintext 1(T_1) – {00 00 00 00 00 00 00 00}
Plaintext 2(T2) – { 01 00 00 00 00 00 00 00}
Key – {CD BF 03 36 9E AD 5E F3}
$T_1 \oplus K$ – {CD BF 03 36 9E AD 5E F3}
$T_2 \oplus K$ – {CC BF 03 36 9E AD 5E F3}
$S(T_1 \oplus K)$ – {7C CB 18 75 57 22 0A F5}
$S(T2 \oplus K)$ – {11 CB 18 75 57 22 0A F5}
$M_1\, S(T_1 \oplus K)$ – {EB 12 90 D9 21 1A 4D E7}
$M_1\, S(T_2 \oplus K)$ – {93 12 90 D9 75 1A 4D E7}
$M_2\, M_1\, S(T_1 \oplus K)$ – {B8 55 8B 3E 22 C3 50 38}
$M_2\, M_1\, S(T_2 \oplus K)$ – {40 F9 93 A8 16 3D 55 C0}.

The first selected plaintext in binary representation consists of only zeros. The second plaintext also consists of zeros, with the exception of the eighth bit. Bitwise addition operation (*XOR*) does not affect the propagation of changes. A change of one bit in the substitution S-box impacts only on one byte, and in the operation *Mixer1* – on every fourth byte. After the above operations, the *Mixer2* operation is performed resulting in the change of the entire ciphertext. Specific numerical characteristics are given below.

The developed algorithm was tested for the avalanche effect. For testing purposes, a random 128-bit plaintext was selected. After the inversion of one bit in each position, 128 new plaintexts were obtained, and all the texts were then encrypted. The probabilities k_i between the obtained ciphertexts and the original one were calculated after each round. The results of the analysis after the first and eighth rounds are given below (Tables IV and V). The average value of ε was 0.07 and 0.062 respectively. The smaller the value of the avalanche parameter, the stronger the avalanche effect is in the transformation.

The generated keys were also tested for the avalanche effect by changing the key bits in the same way as in the plaintext. The results obtained were also positive. The analysis results are represented in Figure 4. As could be seen from Figure 4, the value of k_i is within the interval of (0.4; 0.6).

If the cipher operates with the information presented in binary form, then inverting even one bit in the block of original data will result in independent changing the values of all bits in the corresponding block of encrypted data with the probability of 1/2. It is impossible to break such a cipher in a way less expensive in terms of the number of necessary operations than exhaustive search over the set of possible key values. This condition is mandatory for the cipher of the type in question, which claims to be considered good [13].

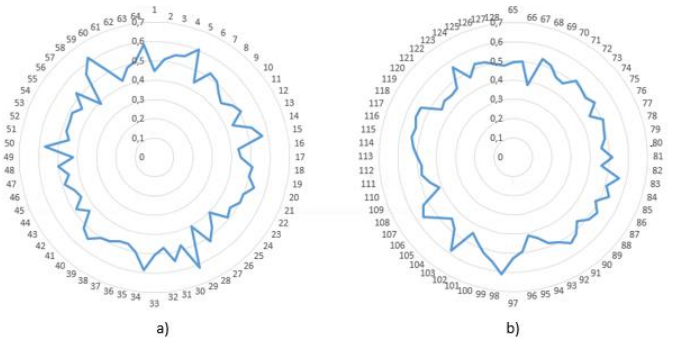

Figure 4. Analysis of the avalanche effect for a key in the full-round algorithm
Diagrams of k_i values for index *i*: a) from 1 to 64, b) from 65 to 128

TABLE IV
ANALYSIS OF THE AVALANCHE EFFECT FOR THE QAMAL ALGORITHM AFTER THE FIRST ROUND

i	k_i	*i*	k_i	*i*	k_i	*i*	k_i
1	0,48	33	0,40	65	0,47	97	0,51
2	0,46	34	0,45	66	0,44	98	0,49
3	0,50	35	0,47	67	0,46	99	0,44
4	0,53	36	0,48	68	0,45	100	0,55
5	0,62	37	0,49	69	0,45	101	0,55
6	0,48	38	0,53	70	0,48	102	0,41
7	0,48	39	0,44	71	0,50	103	0,47
8	0,47	40	0,46	72	0,56	104	0,48
9	0,46	41	0,57	73	0,46	105	0,49
10	0,48	42	0,50	74	0,57	106	0,52
11	0,55	43	0,46	75	0,44	107	0,57
12	0,44	44	0,55	76	0,49	108	0,48
13	0,48	45	0,49	77	0,49	109	0,57
14	0,44	46	0,52	78	0,48	110	0,44
15	0,55	47	0,48	79	0,48	111	0,51
16	0,52	48	0,56	80	0,47	112	0,45
17	0,40	49	0,55	81	0,53	113	0,44
18	0,51	50	0,46	82	0,48	114	0,52
19	0,51	51	0,49	83	0,51	115	0,55
20	0,43	52	0,51	84	0,50	116	0,55
21	0,42	53	0,48	85	0,45	117	0,52
22	0,45	54	0,48	86	0,53	118	0,47
23	0,45	55	0,44	87	0,41	119	0,52
24	0,58	56	0,47	88	0,54	120	0,42
25	0,47	57	0,52	89	0,50	121	0,49
26	0,52	58	0,50	90	0,47	122	0,50
27	0,45	59	0,53	91	0,54	123	0,45
28	0,52	60	0,51	92	0,51	124	0,49
29	0,54	61	0,63	93	0,52	125	0,49
30	0,52	62	0,51	94	0,51	126	0,53
31	0,52	63	0,54	95	0,53	127	0,59
32	0,51	64	0,48	96	0,48	128	0,54

TABLE V
ANALYSIS OF THE AVALANCHE EFFECT FOR THE QAMAL ALGORITHM AFTER THE EIGHTH ROUND

i	k_i	i	k_i	i	k_i	i	k_i
1	0,48	33	0,50	65	0,46	97	0,52
2	0,52	34	0,54	66	0,50	98	0,54
3	0,43	35	0,48	67	0,51	99	0,46
4	0,48	36	0,52	68	0,51	100	0,41
5	0,44	37	0,44	69	0,42	101	0,54
6	0,48	38	0,49	70	0,46	102	0,46
7	0,48	39	0,50	71	0,50	103	0,51
8	0,50	40	0,56	72	0,41	104	0,52
9	0,49	41	0,48	73	0,52	105	0,56
10	0,48	42	0,48	74	0,50	106	0,55
11	0,52	43	0,48	75	0,45	107	0,51
12	0,45	44	0,55	76	0,41	108	0,45
13	0,52	45	0,49	77	0,41	109	0,48
14	0,52	46	0,45	78	0,54	110	0,50
15	0,50	47	0,48	79	0,55	111	0,52
16	0,50	48	0,54	80	0,52	112	0,53
17	0,47	49	0,52	81	0,45	113	0,54
18	0,49	50	0,52	82	0,52	114	0,57
19	0,49	51	0,48	83	0,55	115	0,58
20	0,53	52	0,49	84	0,53	116	0,50
21	0,56	53	0,50	85	0,52	117	0,50
22	0,48	54	0,54	86	0,45	118	0,45
23	0,51	55	0,48	87	0,56	119	0,55
24	0,50	56	0,47	88	0,53	120	0,58
25	0,55	57	0,47	89	0,52	121	0,45
26	0,51	58	0,48	90	0,46	122	0,46
27	0,50	59	0,47	91	0,54	123	0,43
28	0,49	60	0,55	92	0,52	124	0,48
29	0,50	61	0,49	93	0,48	125	0,49
30	0,43	62	0,49	94	0,48	126	0,45
31	0,45	63	0,48	95	0,56	127	0,60
32	0,54	64	0,47	96	0,47	128	0,49

VI. CIPHERTEXT TESTING FOR STATISTICAL SECURITY

In the process of developing ciphers, the task of analyzing their cryptographic properties arises, one of the stages of which is statistical testing. To automate this stage, it is necessary to have a standardized methodology so that the results of statistical testing of various ciphers can be compared [14].

Of particular importance in cryptography is the task of statistical testing of a numerical sequence. To date, there is no single standard set of criteria for evaluating the properties of bit sequences. Various statistical tests evaluate to what extent a bit sequence under consideration is “similar” or “not similar” to a truly random sequence. In each such test, the so-called null hypothesis of the randomness of the sequence is checked (the alternative hypothesis assumes that the sequence is not random). In this case, the significance level α is set, i.e. the probability of a false-negative result and 0.01 or 0.001 is often used as the value for this level. To evaluate the sequence in each statistical test, the so-called P-value is calculated – the probability that an ideal random sequence generator will generate a sequence “less random” than the sequence being studied. The sequence randomness hypothesis is accepted if P-value $\geq \alpha$, otherwise it is rejected.

The methods for assessing the quality of random and pseudo-random sequence generators can be divided into two groups:

1) Graphical tests**.** The properties of sequences are represented in the form of graphical dependencies, by the form of which conclusions are drawn about the proximity of the sequence under consideration to a random one.

The following tests can be attributed to this category: a histogram of the distribution of sequence elements, plane distribution, monotonicity testing, etc.

2) Assessment tests. The statistical properties of sequences are determined by numerical characteristics. Based on the assessment criteria, conclusions are made about the degree of proximity of the properties of the analyzed and truly random sequences. Unlike graphical tests, where the results are interpreted by users with possible differences in their interpretation, assessment tests provide a numerical characteristic that unambiguously determines whether the test is passed or not.

To test sequences for randomness, there are a large number of algorithms, and for convenience, software products containing some test suites have already been implemented. Among them, the most common tests are NIST STS, DIEHARD, CRYPT-X, tests by D. Knuth, and others.

One of the first sets of statistical tests was proposed by D. Knuth in 1969 in his classic work “The Art of Computer Programming”. The tests are based on the χ^2 statistical criterion. The calculated value of the χ^2 statistics is compared with tabular results and a conclusion is drawn about the quality of the sequence. The advantages of these tests are their small number and the existence of fast algorithms for their execution.

A ciphertext, subject to its statistical properties, should not differ from a random sequence. The process of investigation of the statistical properties of ciphertexts comprises the following sequential procedures [5,15,16]:

- Plaintext encryption;
- Execution of the set of statistical tests for the ciphertexts obtained;
- Analysis of the statistical testing results for the ciphertexts;
- Decision on the properties of the ciphertexts obtained.

The computer-based testing was conducted by means of the "Computer-aided system for selecting statistical tests and graphical tests" software package. To investigate statistical properties, graphical and assessment tests were applied.

For the computer-based testing of the algorithm we used:

- 20 files differing by their sizes and extensions;
- 10 different keys.

By using the selected keys and plaintexts we obtained 200 ciphertexts, and then tested them for statistical security. For this purpose, a developed software package was used, which embodied a quality evaluation system for encrypted texts based on graphical and assessment tests.

The results of graphical tests are interpreted by users, so a disparate treatment thereof is possible. Contrastingly, the assessment tests output a specific numerical rating, which makes it possible to uniquely determine if a test has been passed or not.

The histogram of the assessment tests is shown in Figure 5. The results of assessment tests are as follows: the criteria of equidistribution test (frequency test), serial test, serial by character test, gap test, poker test (partition test), coupon collector's test, permutation test, run test, and serial correlation test were met by 95%, 98%, 96%, 95%, 98%, 96%, 98%, 95%, 100% of ciphertexts respectively.

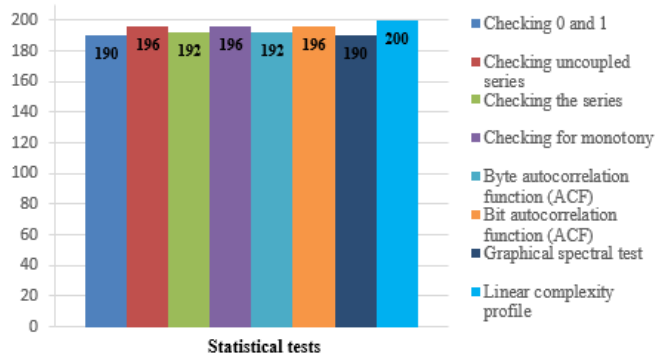

Figure 5. Assessment testing results

VII. Conclusion

Modern block encryption algorithms are subject to the requirements related to their applicability, feasibility, and other factors. The benefits of the developed algorithm are as follows: the algorithm could be effectively implemented in special-purpose hardware, intended for the execution of encryption and decryption operations; it could be easily modified for different levels of security; the transformations used in the algorithm allow for parallel computing (encryption). The results of conducted analyses showed that a minor change in an original message gives rise to a strong change in the encrypted message even with the use of the same key. This cipher property obscures the relationships between the characters of the original text and ciphertext.

From the viewpoint of an adversary, a secure cryptosystem is a black box, input and output information sequences of which are mutually independent, provided that the output ciphered sequence is pseudorandom. Thus, a ciphertext obtained is investigated for pseudo randomness by using statistical tests (testing) and cryptanalytic techniques. The statistical tests showed that the ciphertexts exhibit good statistical properties.

When developing encryption algorithms, it is imperative to analyze them in terms of the strength against different cryptographic attacks. Among the most commonly used at present standard methods are attacks based on the linear and differential cryptanalysis [17-20]. The developed algorithm was investigated against these attacks. As is known, the strength of most algorithms against differential and linear analysis is secured by their S-boxes. This triggered a large number of studies concerning the properties of S-boxes. The algorithm uses pre-developed and investigated S-boxes. The findings are described in [21-23]. The study of the cryptostrength of the algorithm begins with the cryptanalysis of each transformation separately. Then, depending on the results obtained, an analysis of the entire algorithm, i.e. for the whole round transformation, is conducted. The study of the algorithm strength for separate procedures showed good results, which suggest the cryptographic strength of the developed algorithm and the possibility to study the algorithm comprehensively, i.e. considering all transformation procedures and rounds. We continue the work on the security of the algorithm, and the results obtained will be available in the coming papers.

References

[1] W. Stallings, “Cryptography and Network Security: Principles and Practice”, 7th ed.,London: Pearson Education Ltd., 2016.

[2] W. Mao, “Modern Cryptography: Theory and Practice”, Moscow, Williams, 2005, 763 p.

[3] N. Kapalova, A. Haumen, “The model of encryption algorithm based on non-positional polynomial notations and constructed on an SP-network,” *Open Engineering*, vol.8, issue 1, pp. 140-146, 2018.

[4] C.C.Mann, “Homeland Insecurity,” *The Atlantic Monthly*, vol. 290, no.2, September 2002.

[5] M.A. Ivanov, I.V. Chugunkov, “The theory, application and evaluation of the quality of pseudo-random sequence generator,” KUDITS-OBRAZ, Moscow, 2003.

[6] L.K. Babenko, E. A., Ischukova, “Modern Block Encryption Algorithms and Methods of their Analysis,” Moscow, Helios, ARV, 2006, 376 p.

[7] B. Schneier, “Applied Cryptography”, 2nd ed., transl. from English, Moscow: Triumf, 2002.

[8] Ivanov M. A., “Cryptographic Methods of Information Security in Computer Systems and Networks”, Moscow, KUDITS-OBRAZ, 2001.

[9] I. D. Gorbenko, V. Dolgov, R.V. Oleynikov, V.I. Ruzhentsev, M. S. Mikhaylenko, Y. I. Gorbenko, “Development of requirements and design principle of a prospective symmetrical block encryption algorithm,” *News SFU. Engineering science*, vol. 1, no. 76, pp. 183-189, 2007.

[10] C. Shannon, “Works on information theory and cybernetics,” Moscow, IL, 1963, pp. 333-369.

[11] A. Yu. Zubov, “Perfect Ciphers”, Moscow, Gelios ARV, 2003.

[12] R. Hadi, A. Agranovsky, “Practical Cryptography”, Moscow, Solon-Press, 2009p.

[13] I. Vergili, M. D. Yücel, “Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen S-Boxes,” *Turk J Elec Engin*, no. 2, pp. 137–145, 2001.

[14] R. Dragomir, M. Marin, F. Rastoceanu, F. Roman, “Testing block cipher strength with diffusion method,” *The 18th International Conference the Knowledge Based Organization*, 14-16 June 2012, ISBN: 1843-6722, pp. 218-222.

[15] D. Knuth, “The Art of Computer Programming”, Moscow, Williams, 2001, V.3, 600 p.

[16] S.E. Nyssanbayeva, N.A. Kapalova, D.S. Dyusenbayev, K.T. Algazy, “Investigation of statistical properties of a developed pseudorandom sequence generator,” *Proc. of RK MES IICT Sci. Conf. "Modern problems of informatics and computational technologies"*, Almaty, 2018, pp. 210-217, 2018,

[17] Mitsuru Matsui, “Linear Cryptanalysis Method for DES Cipher: Advances in Cryptology,” *Proceedings of Eurocrupt 93, Lecture Notes in Computer Science* 765, pp. 386-397, 1994.

[18] E. Biham, A. Shamir, “Differential cryptoanalysis of DES-like Cryptosystems”, *Extended Abstract, Crypto'90*, Springer-Verlag, pp.2-21, 1998

[19] K. Nyberg, “Differentially uniform mappings for cryptography. I Advances in cryptology,” in *Proceedings of EUROCRYPT'93*, vol.765, Lecture Notes in Compuer Springer-Verlag, Berlin, Heidelberg, New York. pp.55-65, 1994.

[20] V. Korchynskyi, V. Kildishev, O.Riabukha, O.Berdnikov, “The generating random sequences with the increased cryptographic strength,” Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska, vol. 10, no. 1, pp. 20-23, January 2020, https://doi.org/10.35784/iapgos.916

[21] N.A. Kapalova, D.S. Dyusenbayev, K.T. Algazy, “Linear and differential analyses of S-boxes,” *Proceedengs of the 13th International School-Workshop "Problems of optimization of complex systems" under International Conference IEEE SIBIRCON 2017”*, Novosibirsk, September 18-22, 2017.

[22] R.G. Biyashev, M.N. Kalimoldayev, S.E. Nyssanbayeva, N.A.Kapalova, D.S. Dyusenbayev, K.T. Algazy, “Development and analysis of the encryption algorithm in nonpositional polynomial notations,” *Eurasian Journal of Mathematical and Computer Applications*, vol. 6, no.2, pp. 19-33, 2018.

[23] N. Kapalova, D. Dyusenbayev, “Security analysis of an encryption scheme based on nonpositional polynomial notations,” *Open Engineering*, no.6, pp. 250-258, 2016.