CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.centralasianstudies.org

Volume: 03 Issue: 12 | Dec 2022 **ISSN: 2660-5309**

STREAM ENCRYPTION ALGORITHMS AND THE BASIS OF THEIR CREATION

Rahmatullayev Ilhom Raxmatullayebich

Senior teacher, Samarkand branch of Tashkent University of Information Technologies named after Muhammad al-Khwarizmi

Email: ilhom9001@gmail com

Abstract

This article analyzes stream encryption algorithms belonging to the family of symmetric encryption algorithms and their creation bases, as well as types of pseudorandom number generators and their development bases. Pseudo-random generators created on the basis of a systematic-theoretical approach, pseudo-random generators based on the approach based on computational complexity and combinations, and stream encryption algorithms created on their basis are reviewed.

Article history:
Received 6 Oct 2022
Revised form 5 Nov 2022
Accepted 18 Dec 2022

Key words: *Stream (continuous) ciphers, XOR, SVS, OFB, SSL, SET, LFSR, Hardware, Software, Hybrid, combinational generators, filter generators, time control generators.*

A R T I C L E I N F O

----------***----------

Introduction

Stream (continuous) ciphers, unlike block ciphers, encrypt each element of the stream of information, preventing information from being captured in the cryptosystem, and the main achievement is the amount of information, regardless of stream discharge, at a high speed close to the rate of information access in real time. encryption is transmitted without delay.

Stream encryption algorithms belong to the family of symmetric encryption algorithms, where each plaintext symbol becomes a ciphertext symbol depending not only on the key used, but also on its position in the plaintext stream. In stream encryption, the encryption process is based on a different approach compared to block ciphers.

Stream encryption algorithms are gamma-based encryption algorithms, which convert each consecutive 1-bit of the plaintext into ciphertext by XORing it with the corresponding 1-bit gamma key generated by the generator [1].

$$c_i = p_i \oplus k_i \qquad (1)$$

The receiver XORs the ciphertext with the corresponding 1-bit gamma generated by the same encryption generator (using a secret symmetric key) to generate plaintext from the received ciphertext.

$$c_i \oplus k_i = p_i \oplus k_i \oplus k_i = p_i \quad (2)$$

The tolerance of cryptosystems based on stream encryption to various attacks depends on the tolerance of the generator used in the algorithm. And the tolerance of the generator is evaluated by the period of the generated sequence and the degree of randomness. If the generator generates the same sequence in each session or the repetition period is short, it is possible to add the two encrypted ciphertexts by XOR operation and get the XOR sum of the two plaintexts $p_1 \oplus p_2$. The difficulty of deciphering this ciphertext is approximately equal to the difficulty of deciphering a multi-alphabet cipher, which makes the crypto-attack easier.

$$p_1 \oplus k_1 = c_1, p_2 \oplus k_2 = c_2, c_1 \oplus c_2 = p1 \oplus k1 \oplus p_2 \oplus k_2 = p_1 \oplus p_2$$

(3)

Another important characteristic of generators used in stream encryption systems is the degree of randomness of generated sequences [1]. The degree of randomness of blocks of sequences is determined using certain parameters. Generators that develop sequences of pseudo-random numbers with a high level of randomness are an integral part of modern cryptosystems, these sequences are used in cryptography for the following purposes[2]:

- when generating session keys and other keys for symmetric cryptosystems;
- in the generation of initial random values for sufficiently long mathematical quantities used in asymmetric cryptosystems (for example, for the generation of large prime numbers);
- in creating vectors with a high degree of randomness for modes of block encryption algorithms that require a random initial value, such as SVS, OFB;
- when generating random values for long parameters used in electronic digital signature algorithms;
- generation of random values in protocols such as SSL and SET, which are required to send the same data in different ways through the same protocol, etc.

Main part

The problem of generating a random sequence with an arbitrary probability distribution regularity ultimately boils down to the problem of generating a uniformly distributed sequence. In uniformly distributed sequences, for an arbitrary random value $t \in N$, the discrete probability of an element in the set of sequences $x_t \in A$ is equal to $P\{x_t, A\} = 1/N$ [2]. If the squared differences of the probabilities of each element in this set of sequences A lie between 0.05 and 0.95, this sequence can be considered a random sequence.

According to the property of uniformly distributed sequences, if $A(a_i)$ is a uniformly distributed random sequence and $V(b_i)$ is a uniformly distributed and non-random sequence, then $S(s_i) = A(a_i) \oplus V(b_i)$ - the resulting sequence will be a uniformly distributed random sequence. This property can be used to combine algorithms.

Uniformly distributed random sequences are divided into pseudorandom sequences and true random sequences. Such sequences can be developed in 2 different ways[1]:

- through physical generators;
- through software generators.

A sequence generated by physical generators is a truly random sequence, such a sequence is generated only once, and there is no possibility of its subsequent generation in the same form with any regularity. Therefore, keys generated in physical generators cannot be used in stream encryption.

Sequences generated by software generators are called pseudorandom sequences, and these sequences

 ISSN: 2660-5309

can be recreated in the same form and of sufficient length using the key used to generate them.

Stream encryption systems use only pseudo-random and uniformly distributed random sequence generators to speed up the process of encryption and decryption. The generators that generate even distributed sequences and the stream encryption algorithms based on them were created based on certain approaches.

Stream encryption algorithms based on software generators generating pseudo-random sequences are mainly created based on the following approaches[4,5]:

1. Algorithms developed on the basis of pseudo-random generators created on the basis of a systematic-theoretical approach;
2. Algorithms created on the basis of pseudorandom generators based on the approach based on computational complexity;
3. Algorithms created on the basis of pseudo-random generators based on combination.

Creating stream encryption algorithms based on a system-theoretical approach is similar to creating block encryption algorithms in many ways. The cryptoresistance of stream encryption algorithms created based on this approach is equated to the difficulty of the problem, which is complex, taking into account the fundamental mathematical criteria and laws, and the solution method is considered unknown or non-existent. Algorithms are created on the basis of the theoretical achievements of mathematics, which generate sequences with a sufficiently large period length, evenly distributed blocks, and non-linearity. Then the tolerance of the created algorithm to various cryptanalysis methods is evaluated. If the created algorithm is resistant to existing cryptanalysis methods and the generated sequences meet the requirements of randomness, a positive conclusion is given that this algorithm can be used in practice.

The originally created stream encryption algorithms were also developed based on a systematic-theoretical approach.

There are the following requirements for streaming encryption algorithms based on a systematic-theoretical approach[1,5]:

- algorithm-based pseudo-random sequence generator generates sequences with a sufficiently long period;
- high degree of non-linear complexity of the generator;
- blocks of generated pseudo-random sequences should have a flat statistical distribution indicator;
- gamma elements (bit, byte, partial blocks) of a pseudo-random sequence are formed by the influence of all other elements, that is, they have the property of effective mixing;
- abrupt change of gamma elements of the pseudo-random sequence, that is, having the property of effective propagation;
- these functions of the algorithm reflections should satisfy the condition of non-linearity and ensure that they give rapid effect (avalanche effect).

The difficulty of proving the reliability or robustness of algorithms can be seen as a general shortcoming of stream encryption algorithms created on the basis of a system-theoretic approach.

Based on the basis of their creation, the generators in stream encryption algorithms created on the basis of a systematic-theoretical approach can be divided into elementary recurrences, shift registers, one-way functions, and generators based on independent replacement of bytes and blocks of bits.

A computational complexity approach is based on hard-to-solve math problems. Currently, it is possible to point out the problems of dividing large numbers into prime multipliers, discrete logarithms, solving systems of linear equations of sufficiently high order in finite fields, and solving complexities related to elliptic curves as difficult problems of mathematics. In fact, these problems are theoretically solvable and can be successfully solved using computer systems. However, for certain large parameters, the resources (calculation and time resources) required for the solution of the relevant problem are considered to be difficult problems to solve due to the fact that they exceed the level of available resources [6].

 ISSN: 2660-5309

The practical robustness of stream encryption algorithms based on a theoretical approach to computational complexity is proven by equating the difficulty of the above-mentioned hard-to-solve mathematics problems. Generators in complexity-based algorithms are difficult to create in software or hardware. Since such stream encryption algorithms use very large numbers, complex operations such as multiplication and exponentiation are used, implementation in hardware and software becomes complicated. Since the encryption and decryption process in these algorithms is slow, these algorithms cannot be used for speed and time-sensitive information transmission (voice, video). It is advisable to use such algorithms for transferring small amounts of information with a high level of confidentiality, for example, encryption keys of symmetric block encryption algorithms.

The combined theoretical approach is a method of creating new algorithms based on the combination of algorithms developed on the basis of system-theoretical and complexity-based approaches.

In this approach, a new algorithm is created based on the combination (unification) of algorithms (reflections) that generate existing pseudo-random sequences. The robustness of this algorithm depends on the complexity of each of the reflections and algorithms used in it.

Creation of pseudo-random sequence generators based on combination is carried out by combination of algorithms with random parameters, polynomial combination, McLaren-Marsali methods.

Most of the crypto-tolerant stream encryption algorithms based on shift registers used up to now are created by polynomial combination of shift registers.

Most of the stream encryption algorithms in widespread use today are based on shift registers, that is, linear feedback shift registers. These shift registers are also called Galois registers or Fibonacci registers. The following reasons can be given for the successful use of this type of stream encryption algorithms[1,3].

1. The statistical characteristics of sequences generated using pseudo-random number generators based on inversely connected shift registers are good.
2. Analyzing the characteristics of this type of generators is easy compared to other generators.

Inverse feedback shift registers are divided into linear feedback shift registers and nonlinear feedback shift registers. The general scheme of shift registers is shown in Figure 1.

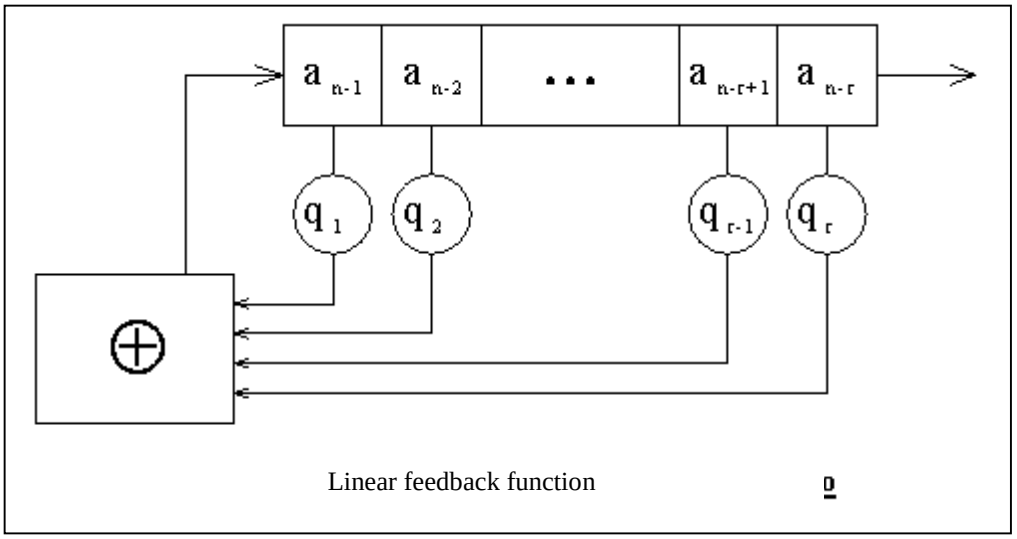

Figure 1. Overview of an inverse-coupled shift register

Generators created on the basis of shift registers consist of a shift register and a feedback function. In the process of software and hardware implementation of algorithms developed on the basis of generators based on shift registers, the number of shift registers is selected equal to the number of registers of the microprocessor in order to ensure fast operation. Since the majority of microprocessors currently work with 64-bit registers, it is desirable to make the length of shift registers equal to 64 bits in the software. Then it is ensured that the period

 ISSN: 2660-5309

of the sequence created on the basis of correctly selected parameters reaches the maximum, that is, 264 bits. Another part of the shift registers is the feedback function (Figure 2). The inverse function adds the values of the bits in the polynomial positions of the register with XOR reflection at each clock cycle, and enters the resulting value by shifting the register's most significant digit. The smallest discharge value is transmitted to gamma.

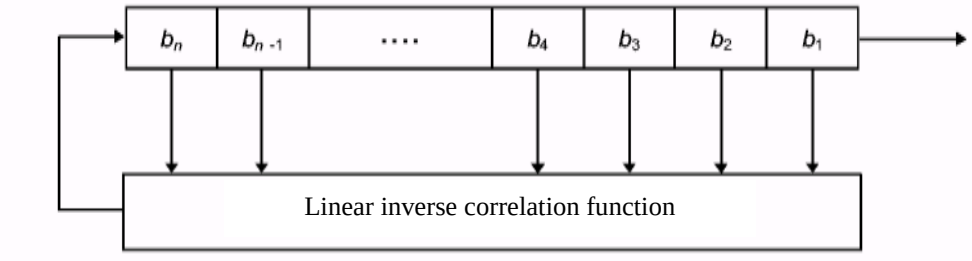

Figure 2. Linear feedback shift register

One of the linear feedback shift registers is the Galois configuration (Figure 3). In the Galois configuration, the bit value transmitted to the gamma is involved in the inverse coupling function. The output bit is XOR to each bit of the register and is given by shifting the high bit of the register. The least significant bit value is passed to gamma and used in the inverse correlation function. In order for the period of the sequences leaving the register to be maximal, the arguments of the inverse connection function should be taken from the terms of the non-deducible polynomial generator of the register.

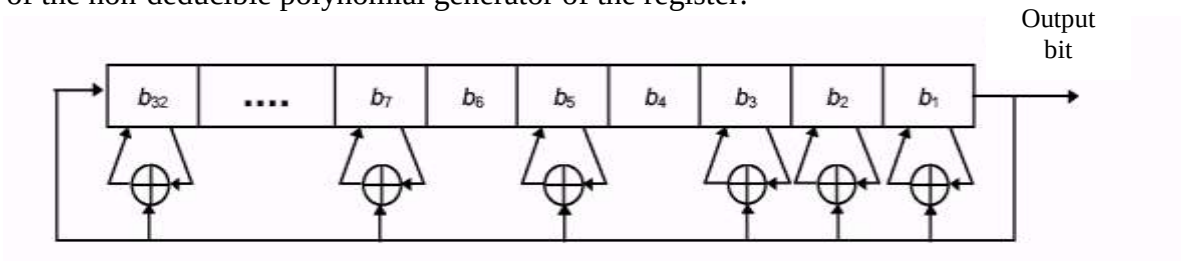

Figure 3. A shift register based on the Galois configuration

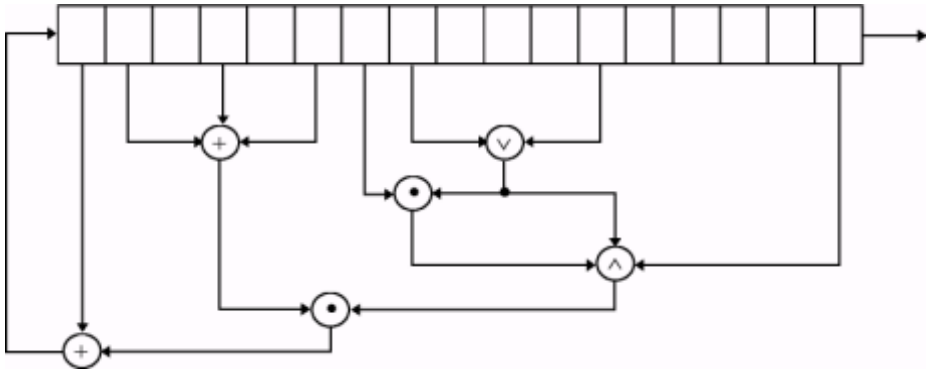

Figure 4. Nonlinear feedback shift register

In nonlinear feedback shift registers, the feedback function is implemented by using several different nonlinear reflections. XOR, AND, OR logical operations are used in the inverse connection function presented in Figure 4. However, mathematical methods that adequately analyze sequences generated by generators based on nonlinear shift registers have not yet been developed. Therefore, the following problems can be shown in generators implemented by nonlinear feedback registers:

- generated pseudo-random sequences may deviate from the characteristics of a flat distribution, that is, the number of "0"s and "1s" may not be equal;
- the sequence period may be shorter than expected;

 ISSN: 2660-5309

- the period of the sequence can be different for different values of the initial values, that is, the period of the sequence generated by the generator for any arbitrary initial value is maximum when the parameters that meet certain requirements are selected may not;
- the initially generated sequence may appear to be random, but after a certain state of the register is reached, the generated sequence may consist only of "0" or "1".

The directions of creation of pseudo-random generators, which are the basis of existing stream encryption algorithms, can be shown in general as follows (Fig. 5).

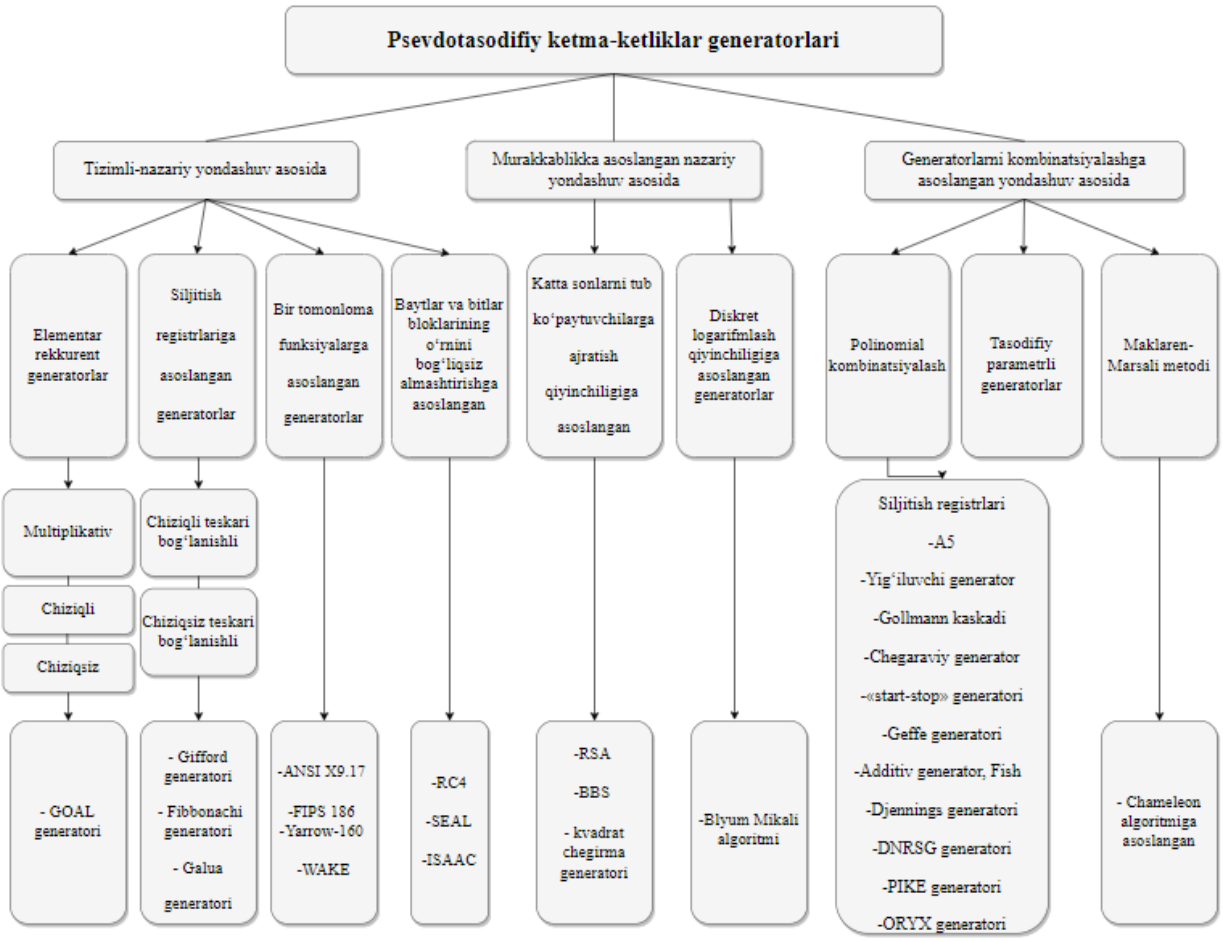

Figure 5. Directions for creating pseudorandom generators

Compared to block ciphers, there is no standard model for developing continuous ciphers, prompting cryptographers to develop a number of stream cipher models. According to the purposes of practical use (implementation), stream ciphers are divided into several categories, and these categories include stream ciphers with special properties. There are 3 main directions of these categories (Fig. 6) [7]:

– Hardware stream ciphers;
– Software stream ciphers;
– Mixed (Hybrid) stream ciphers.

 ISSN: 2660-5309

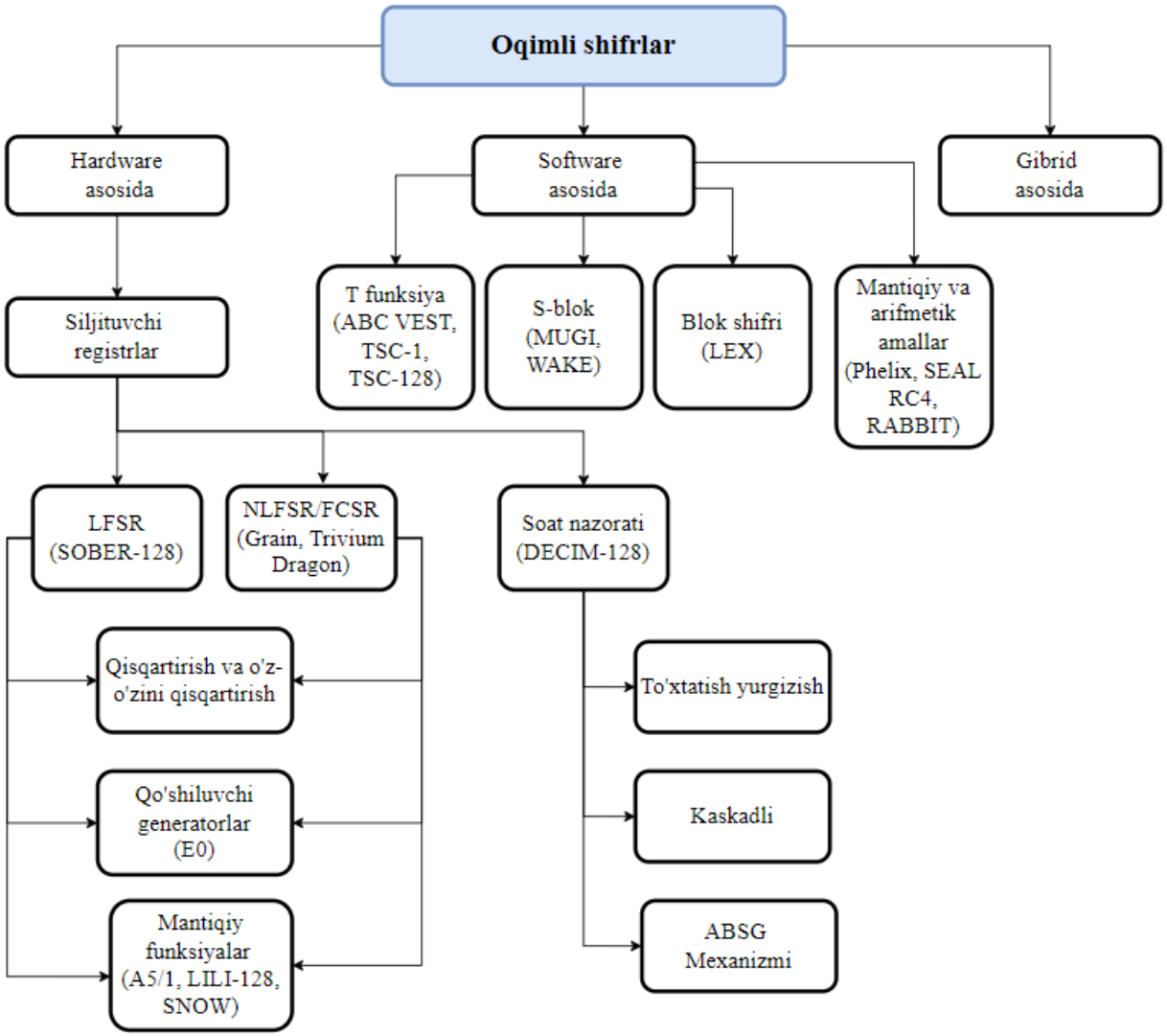

Figure 6. Classification of stream ciphers according to their practical application

The classification of hardware-based stream ciphers includes stream ciphers based on FSSR/NLFR, clock control, and LFSR. The use of hardware stream ciphers plays an important role in the security of many cryptographic applications. Modern algorithms such as DECIM v2, Edon-80, F-FCSR-H v2, Grain v1, MICKEY v2, MOUSTIQUE, POMARANCH v3, Tvirium are examples of the category of algorithms developed in the direction of hardware.

Software-based stream ciphers include T-function, block cipher, S-block, and simple logical and arithmetic operations. Compared to hardware-based stream ciphers, ciphers of this category differ in that they are based on bit manipulation (substitution, replacement) and appear logical. Modern algorithms such as CryptMT v3, DRAGON, HC-128, LEX v2, NLS v2, Rabbit, Salsa20, SOSEMANUK can be cited as examples of software-based stream ciphers.

Hybrid-based stream ciphers consist of stream ciphers created based on a combination of hardware and software. The majority of stream ciphers of this category are based on LFSR.

Generator functions serve to generate a sequence of numbers by performing certain reflections on given initial values. There are following types of generator functions:

- combined generators;
- filtering generators;

 ISSN: 2660-5309

- time control generators.

Combinational generators are built by combining (combining) several inversely connected shift registers (Fig. 7, where: *f* is a combination function).

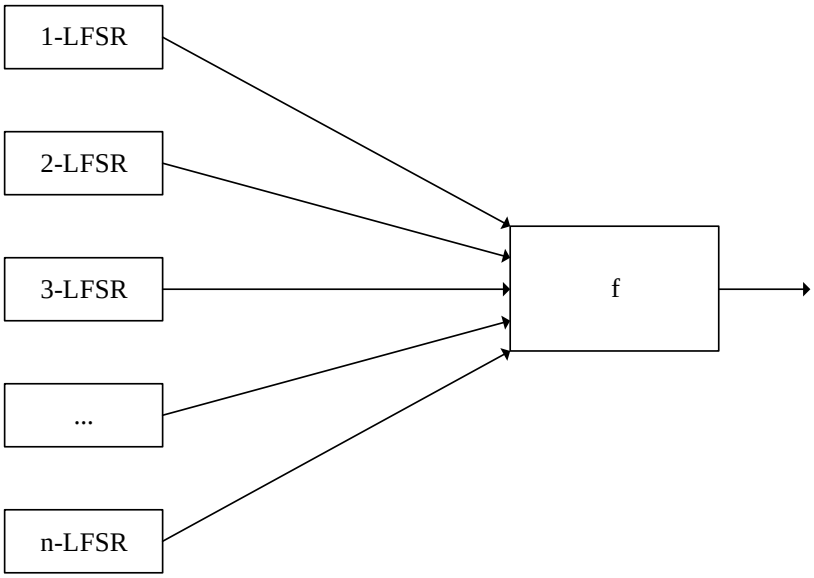

Figure 7. Combined generators

In filtering generators, a single inversely coupled shift register is used (Fig. 8, where: *f* is a filtering function).

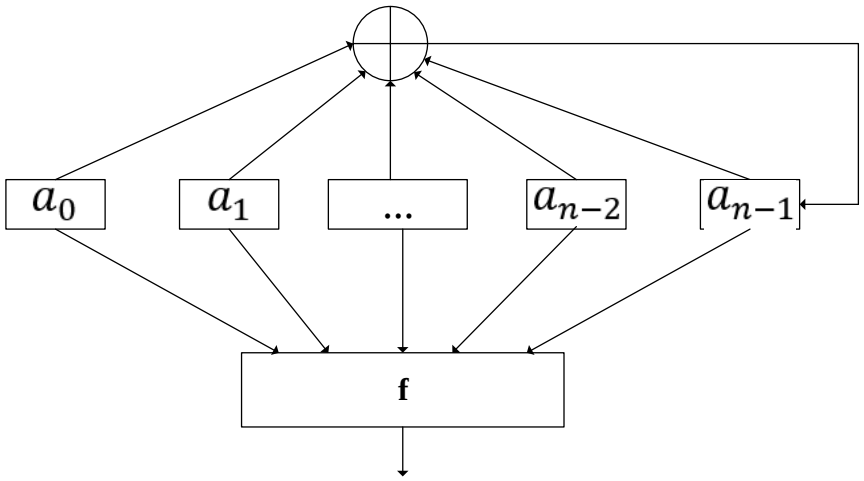

Figure 8. Filter generator

Timing generators use multiple inversely coupled shift registers like combinational generators, only the values of the registers are dependent on each other.

Conclusion

According to the analysis of stream encryption algorithms, in contrast to block encryption algorithms, despite the fact that many methods and directions for creating crypto-resistant continuous encryption algorithms have been developed in this field, there are no single methods that express their commonality with each other [2].

Generators that develop sequences of pseudorandom numbers are an integral part of continuous encryption systems, and the tolerance of this system depends on the tolerance of these generators. The tolerances of stream encryption algorithms lie between the tolerances of single-letter permutation algorithms and the one-time notepad algorithm tolerances.

The sequences generated by the pseudo-random number sequence generator look like true random

 ISSN: 2660-5309

sequences, but this sequence can be recreated using just such a generator and the key used in it. This property ensures efficient application of stream encryption algorithms in practice and allows to reach the tolerance level of the cryptosystem up to the tolerance of the encryption algorithm using a one-time pad.

Further research aims to consider cryptanalysis methods used in the evaluation of stream encryption algorithms.

References

1. Schneier B. Applied cryptography. Protocols, algorithms, source texts in C language. - M., Ed. TRIUMPH, 2003. - 816 p.
2. Kharin Yu.S., Bernik V.I., Matveev G.V., Agievich S.V. Mathematical and computer bases of cryptology: Textbook. - Minsk, LLC "New Knowledge", 2003. - 382 p.
3. Asoskov A.V., Ivanov M.A. Stream ciphers, M: Kudits-Obraz, 2003. - 336 p.
4. Akbarov D.Ye. Cryptographic methods of information security and their application. - Tashkent, "Mark of Uzbekistan" publishing house, 2009. - 432 p.
5. http://www.cryptography.ru
6. Musayev A.I. Research the basics of existing stream encryption algorithms and create new crypto-resistant algorithms. Master's dissertation in the field of information security. Tashkent, 2008. - 81 p.
7. Suwais K., Samsudin A. New Classification of Existing Stream Ciphers. Universiti Sains Malaysia(USM), Malaysia 2010.

 ISSN: 2660-5309