The Technique for Development of Encryption Algorithms with Improved Cryptographic Strength

Dmitriy Shatokhin
Karaganda State Technical University
Karaganda, Republic of Kazakhstan
dvsh68@mail.ru

***Abstract*— One of the main objectives in development of cryptographic algorithms is to ensure their cryptanalysis resistance. However, cryptanalysis methods are constantly improving, and due to this fact new and more complex cryptographic algorithms that can withstand new methods of cryptanalysis are being developed. This paper describes the original technique for creating cryptographic algorithms with fundamentally improved cryptographic strength. It also gives a brief technical description of the cryptographic algorithm created with the help of such technique.**

I. Introduction

As you know, one of the fundamental principles of cryptanalysis is that cryptosystem security is based entirely on the security of the encryption key. It means that detailed knowledge of the cryptographic algorithm by a cryptanalyst must by no means affect the cryptosystem security [1]. However, it should be noted that the detailed knowledge of the cryptographic algorithm is also one of the key factors of successful cryptanalysis. It follows that if in any way one limits the knowledge of the cryptanalyst about at least one significant part of the algorithm being used, the cryptanalysis of such a cryptographic algorithm will be quite difficult as well.

It is to be noted that, despite the evidence of such approach, public media sources provide not a single cryptographic algorithm that would fully implement a similar technique.

To describe this idea in a more detailed way, first of all, it is necessary to consider the concept of the main function of a cryptographic algorithm.

II. The technique description

A. The main function of a cryptographic algorithm

The main function of cryptographic algorithm shall mean the key cryptographic transformation process that is used directly to carry out encryption or decryption. Any of the existing cryptographic algorithms incorporates such a function, which is fed to the input with a plaintext and a key (or its part), and the output is represented by the result of cryptographic transformation. For example, in block cryptographic algorithms, it is a round function executed repeatedly in every round. In synchronous stream cryptographic algorithms, it is a selection function or a function of calculating the next pseudo-random bit or byte. The complexity and nonlinearity of the key function directly affects the complexity of the algorithm cryptanalysis.

If, however, a cryptographic algorithm contains not one but several functions that will change one another according to a certain schedule during the algorithm work, cryptanalysis of such an algorithm will be complicated many times over. To implement this idea, a special cryptographic algorithm development technique has been created and abbreviated as VOMF (**VO**latile **M**ain **F**unction).

B. Volatile main function and VOMF technique

Thus, to complicate the task of cryptanalysis, one should do the following: during the encryption, the algorithm should from time to time change the basic function according to a certain schedule throughout the process of encoding, and these changes of the key function should be of random and unpredictable nature for a cryptanalyst.

With this approach, the only way to hide the principle of key function change from a cryptanalyst is to put all the information about this into the encryption key. This, of course, will lead to some increase in the size of the key. However, the increasing size of the key is not considered to be a very significant drawback, taking into account potential benefits of such a method. Thus, the encryption key can contain all the information about the change of encryption functions and their performance sequence.

The practical implementation of this technique during development of cryptographic algorithm is as follows:

1) When creating a cryptographic algorithm, it is necessary to develop several interchangeable functions performing basic cryptographic transformation. The number of these functions is not limited, but its increase, on the one hand, increases the resistance of the algorithm, and increases the memory requirements on the other hand. It is also necessary to ensure that the performance speed of all of these functions is about equal.
2) It is essential to create an auxiliary algorithm which, with the use of part of the encryption key, will in a certain way create a schedule for the work of these interchangeable functions and their replacement sequence. For example, for this purpose it is possible to use a pseudo-random sequence generator, or another mechanism.
3) At the start of the cryptographic algorithm work, firstly

a key schedule algorithm is implemented, and then a schedule of interchangeable functions work is made.
4) It is also necessary to provide a mechanism that determines whether all interchangeable functions will take part in this session of the cryptographic algorithm or not all of them.
5) Further on, the encryption algorithm works as usual, except that the interchangeable functions are performed each time in accordance with their created work schedule.

Thus, the encryption key of such a cryptographic algorithm includes additional information necessary for creation of the schedule of such interchangeable functions work. Cryptanalysis of such an algorithm is considerably hampered due to the fact that the cryptanalyst has no information on the schedule of interchangeable functions.

The next, most interesting step in the further progress of this technique is the creation of a mechanism for dynamic generation of interchangeable functions. With this approach, the interchangeable functions are not defined initially, but they are dynamically formed during the execution of each iteration of the algorithm. In this case, the cryptanalyst will have considerably less information about such interchangeable functions, since the set of all possible functions is very large. At present time work in this direction is going on.

There are also two disadvantages of using such a technique of cryptographic algorithms development. The first one is the increase of the key size, which generally is not a serious drawback. The second one is slowing down of the algorithm work due to the complication of its structure. The best results for this technique were obtained during the development of stream type cryptographic algorithms.

C. IMPASE cryptographic algorithm

IMPASE cryptographic algorithm (**IMP**roved **A**lgorithm of **S**tream **E**ncryption) is an algorithm of synchronous stream encryption that is based on the above VOMF technique.

Being synchronous byte-oriented stream cryptographic algorithm, it actually represents a cryptographically strong pseudo-random gamma sequence generator the sequence of which is superimposed byte by byte on the plaintext of module 2 during encryption and decryption operations. Thus, the cryptographic strength of the algorithm is determined entirely by the strength of generated gamma sequence.

The operating principle of this cryptographic algorithm may be briefly described as follows.

The core of the cryptographic algorithm is *N* of 256-byte arrays (in the basic version of the algorithm, *N* varies from 4 to 16, depending on the combination of encryption key and initialization vector), 6 interchangeable functions of selecting bytes from arrays as well as table of parameters for dynamic determination of the function for selection bytes from arrays. At the initial stage, in the key schedule algorithm, initialization vector with a special algorithm is combined with the encryption key using special algorithm [2]; then the arrays are filled with pseudo-random data, depending on the key value. After that, the tables of parameters are filled out with the values depending on the key, that are later used to dynamically determine the functions of bytes sample from *N* arrays, as well as post-processing of these arrays. Next, gamma sequence itself is created. Each iteration of the algorithm generates one 32-bit word. To do this, with the help of tables of parameters, the selecting function of 4 bytes from arrays is determined, and these bytes concatenated into a 32-bit word that is sent to the output stream. Next, the function of post-processing is determined, which is used to swap bytes within the array (the number and type of swaps as well are determined by the values in the table of parameters).

The change of at least one bit of encryption key or initialization vector results in a significant change in the condition of the algorithm core and as a result leads to a complete change of gamma sequence.

The implementation of VOMF technique in this cryptographic algorithm consists in unpredictability of function of bytes selecting from the arrays and the unpredictability of function of bytes swap in arrays; these functions change with each iteration, and are determined by the table of parameters dynamically. The tables of parameters are also modified at times.

The use of this technique, in addition to the cryptographic strength, provides a huge period of cryptographic pseudorandom sequence generated: the value of lower limit of the generated sequence period is more than 10^{2000} of 32-bit words. This, in turn, makes it possible to use this cryptographic algorithm in any communication channels without risk of repeating cryptographically resistant gamma sequence even with prolonged use of the same encryption key.

The main characteristics of the cryptographic algorithm are as follows:

1) Encryption key length is from 256 bits to 512 (preferably 512 bits).
2) The length of the initialization vector used is 64 bits.
3) The encryption speed in software implementation is more than 300 Mbit / s (on the Intel Pentium Core i3 processor, 3.3 GHz, OS MS-DOS kernel).
4) The use of memory for the algorithm core and internal variables is no more than 6 kb (with the possibility of flexible adjustment of the amount of memory usage). The minimum required amount of memory is about 1.5 Kb.
5) A huge period of generated sequence.

This cryptographic algorithm is relatively easily implemented in software and can be used in all applications, operating systems, communication systems and specialized controllers.

III. CONCLUSION

This paper describes the special technique for developing an encryption algorithms with improved cryptographic strength, as well as a brief description of stream cryptographic

algorithm created with the help of this technique. For this moment the work on further progress of discussed technique now is going on. We believe that the use of the described approach will allow to create more secure cryptosystem.

REFERENCES

[1] B. Schneier, *Applied cryptography*. Second edition. John Wiley & Sons. 1996.

[2] B. Schneier, "One-Way Hash Functions", Dr. Dobb's journal, v. 16, n. 9, Sep 1991.