Modernized RC4 encryption algorithm

V Semenov, S Makov, I Grinev
Don State Technical University, Gagarin square 1, Rostov-on-Don 344000, Russia
E-mail: vvsemenov@mail.ru

Abstract. A modernized RC4 encryption algorithm with implementation of the resulting algorithm in program for encrypting files is proposed. The program has simple and intuitive interface. Testing showed that the modernized encryption algorithm is slower for 4 milliseconds than the standard RC4, which is explained by a more complex key expansion algorithm. Proposed algorithm can replace RC4 without substantial loss in file encryption.

1. Introduction
Advent of new information technologies and development of powerful computer systems for storing and processing information have increased the levels of information security and necessitated that effectiveness of information security grows along with complexity of data storage architecture. Protection of economic information becomes necessary: documents are developed to protect information; recommendations on information protection are formed, etc [1].

Nowadays, a wide range of information storage and processing systems is existed, where the information security factor is taken into account in the process of their design. Such information systems include, for example, banking or law systems for safe document management and other information systems for which the protection of information is very important [2].

Designing secure systems is impossible without use of various data encryption algorithms. Different algorithms approach different areas. There is no one universal encryption algorithm. The cryptographic strength of any algorithm is examined by the time and the number of attacks undertaken on it to crack it. More unsuccessful attempts mean more stability working of this algorithm [5-8].

Relevance of using modern encryption algorithms in information systems is that it increases security of information storage and transmission. The purpose of this work is to develop an encryption algorithm based on standard RC4 with the performance of a standard RC4 encryption algorithm and eliminating its vulnerabilities, also implementation of the modernized algorithm as a file encryption program.

In 2001, Fluhrer, Mantin and Shamir published a paper on vulnerability of RC4 key schedule algorithm [5]. They showed that among all possible keys, first few bytes of key stream are completely nonrandom. From these bytes, can be obtained information about the key used by the cipher, which substantial reduces the reliability of the algorithm. RC4 is a stream cipher and is widely used in various information protection systems in computer systems. Main features of the cipher are high speed and variable key size. A typical implementation performs 19 machine instructions per byte text. Like any stream cipher the RC4 algorithm is constructed on the basis of a uniformly distributed pseudo-random bit generator parameterized by the key.

This paper describes a possible method of algorithm modernization for eliminating the identified vulnerability.
Figure 1. The “wet” winding method

The squeezing rollers 8 remove the excess of binder. The electric drive 9 changes the gap between the rollers. The tension device 10 consists of a fixed roller 11 and a roller 12 that moves along the tensioner using an electric drive 13. By this way, the “wet” tape coverage angle changes. The measurer 14 measures the tape tension. The roller 15 lays the tape on the product 16 along a special path. For this purpose, it is equipped with stacking mechanisms, which is not shown on the figure. The drive 17 rotates the product 16. Several interconnected systems control the winding.

The change of the movable roller 12 position maintains the winding tension at a predetermined level.

The improvement of the winding technology imposes stringent requirements for the drive speed and their adaptation to changing operation modes. This article is devoted to the development of such a system. The experience of using the tension control systems allowed us to determine the basic requirements for the developed systems. In particular, the transition time must not exceed the period of 0.1 second and the excessive correction must not be more than 10%.

2. RC4 algorithm
Consider the operation of the RC4 algorithm. The RC4 algorithm consists of two parts [3-4]:
1. Key creation (also called key extension).
2. Encryption.

The key in RC4 is a sequence of bytes of arbitrary length, which is used to build the initial state of the cipher $S$ - permutation of all 256 bytes. (Fig. 1, a).

Initially, $S$ is filled with sequential values from $0 \ldots 255$, and $K$ is filled with a key (if necessary, the key is repeated to fill the entire array). Then each element $S_i$ swaps with an element with $i$ number, which is determined by the key element $K$, the element itself and the sum of the numbers of elements exchanged at previous iterations, i.e. $j = S_i + K_i + j$. The values of the counters $i$ and $j$ are initially equal to 0. (Fig. 1, b).

The following actions are performed for encryption. The next element of the pseudo-random permutation $S_i$ of all bytes is swapped with the element $S_j$, where $i = i + 1 \text{mod} 256$, and $j = j + S_i \text{mod} 256$. As the next byte, the value of third element $S$ is given, whose number is equal to the sum $S_i$ and $S_j$. The value of counter $i$ is initially 0, but it increases by 1 already before the first sample $S_i$. The value of $j$ is also initially 0.
3. **Modernized RC4 algorithm**

A lot of researches for ferrous metallurgy, pulp and paper, textile, chemical and electrical industries deal with the mathematical description of elastic tape. Differential equations obtained by a number of authors [2–8] describe the behavior of textile materials, metal tape and ropes when their winding or rewinding. There is a mathematical description of the "dry" and "wet" elastic tape [9–14].

The modernization of the encryption algorithm consists in changing the algorithm of key expansion and byte encryption.

Initially, the KArray is filled with sequentially entered key from 0 to 255. If the key is less than 256 characters, then it is repeated until the entire array is full. Next, KArray is split into two equal arrays K1Array and K2Array. Array K1Array is filled with elements using the formula

\[ K1Array[k] = KArray[(i0 + iMixing) \mod l] \]

array K2Array is filled with elements using the formula

\[ K2Array[r] = KArray[i0 \mod 64] \].

Next, K3Array and K4Array arrays are filled. Array K3Array is filled with elements using the formula:

\[ K3Array = 2 \cdot K1Array[i] + K2Array[i] \mod 256 \]

Array K4Array is filled with elements using the formula:

\[ K4Array = K1Array[i] + K2Array[i] \mod 64 \].

Then K3Array array and K4Array array are combined into a single SessionArray array. Elements from K4Array are put on odd positions and elements from K3Array are put on even position in the SessionArray array. Each element is superimposed with the serial number of this element in the SessionArray using bitwise logical operation XOR.

Theoretical part

The following expression shows the winding tension [12, 13]:

\[ x = x + 1 \mod 256 \]

\[ y = Sx \mod 256 \]

\[ p = Sx \]

\[ Sx = S[y] \]

\[ S[y] = p \]

\[ t = (Sx + S[y]) \mod 256 \]

\[ Result = bt \ xor S[t] \]

**Figure 2.** RC4 encryption algorithm
The actions described above are repeated Mixing times until the key is finally formed. If the Mixing value is not reached, then the values from SessionArray are transferred to KArray using the formula $KArray[j_0] = j_0 + SessionArray[j_0] \mod Mixing$. The value of Mixing is calculated using XOR operation of superimposing the elements of the user key. If received value is less than 32, then Mixing is equal to 32 (Fig. 3). The byte encryption process occurs as in the standard algorithm (Fig. 2, b). The initial value of the counters is calculated by the formulas: $i = Mixing \mod l$ and $j = (Mixing + l) \mod 64$.

As a result of modernized the algorithm, the previous vulnerability is fixed, while the speed of the algorithm does not change substantial. The algorithm can be used in commercial applications for data protection, since the software implementation of the proposed modified algorithm does not present substantial difficulties.

**Figure 3.** Flowchart modernized RC4 encryption algorithm
The appearance of the program is shown in Fig. 4

![Image of data encryption program]

Figure 4. Appearance of data encryption program

Where, $\Delta M_d$ is the tensioner drive moment, Hm; $\Delta \Omega_d$ is the motor shaft rotation frequency, rad/s; $J_\alpha$ is the equivalent moment of inertia, kg/m²; $k_7$ is the reduction gear ratio.

We use the following formulas to calculate the coefficients:

$$
\begin{align*}
  k_1 &= -\left( \frac{\partial F}{\partial a} \right)_0; \\
  k_2 &= -\left( \frac{\partial F}{\partial v_2} \right)_0; \\
  k_3 &= -\left( \frac{\partial F}{\partial S_\alpha} \right)_0; \\
  k_4 &= -\left( \frac{\partial F}{\partial S_1} \right)_0.
\end{align*}
$$

(1)

The program [9] has a simple and intuitive interface, which greatly facilitates the work of unprepared user. Possible errors that user can make are processed by the program. Also, the program displays a message describing error.

One of the tasks in developing the program was to maintain speed of encryption and decryption of files, regardless of algorithm used. To examination the speed of the program was taken a 1 MB file which was encrypted and then decrypted by two encryption algorithms. The results are shown in table 1.

### Table 1. Runtime in milliseconds

| Algorithm     | Encrypted | Decrypted |
|---------------|-----------|-----------|
| RC4           | 172       | 172       |
| Modernized RC4| 176       | 176       |

As can be seen from table 1, the modernized encryption algorithm is slower for 4 milliseconds than the standard RC4, which is explained by a more complex key expansion algorithm. Therefore, this algorithm can be used to encrypt a continuous data stream.

Security of the encryption algorithm used depends on the key length and the cryptographic strength of the algorithm itself. Users exchanging ciphertexts between themselves must exchange keys so that a potential attacker could not intercept the key.

### 4. Conclusions

In this paper, we propose the modernization of the RC4 encryption algorithm, which allows us to fixed the existing vulnerability while maintaining the data encryption speed.

### References

[1] Subashini S, Kavitha V, Journal of Network and Computer Applications, 34 (1) (2011) 1-11

[2] Mister S, Tavares S, Proceedings of SAC 1998, LNCS, 1556 (1999)
[3] Prasithsangaree P, Krishnamurthy P, Telecommunications Program University of Pittsburgh, Global Telecommunications Conference, (2003) 1445-1449
[4] Mister S, Tavares S, Selected Areas in Cryptography (Kingston, ON, 1998), 1556 (1999) 121–143
[5] Wenbo Mao. Modern Cryptography: Theory and Practice, (Prentice Hall, 2003)
[6] Paul S, Preneel B, Fast Software Encryption 2004, 3017 (2004). 245–259
[7] Ferguson N, Schneier B, Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. (Wiley, 2003)
[8] Smart N. Cryptography: An Introduction. (McGraw-Hill College, 2004)
[9] Semenov V V., Kryuchkov I M, Modernized encryption algorithm RC4. Certificate of state registration of computer programs №2012616992 (03.08.2012).