Computer Science Journal of Moldova, vol.**3**, no.3(9), 1995

Fast software encryption system based on local pseudorandomness

A.Moldovyan N.Moldovyan

Abstract

New software-oriented single key cryptosystem based on local pseudorandomness is considered. Its cryptorobustness is characterized in probabilistic terms. The minimal size of the known (chosen) plaintext has been estimated to be $> 3 \cdot 10^{31}$ ($> 10^{21}$) bytes the cryptorobustness being $> 10^{57}$ ($> 10^{37}$) operations.

KEY WORDS: software-oriented cryptosystem, encryption, cipher, local pseudorandomness

1 Introduction

Elaboration of the soft information protection systems or of the computer security software is connected with the problem of designing effective high-speed soft cryptomodules. It is difficult to solve this problem on the basis of known cryptosystems. It is due mainly to the fact that the greatest progress has been achieved in developing hardware-oriented cryptalgorithms, such as DES [1], Russian standard No 28147-89, FEAL-N [2], REDOC II [3].

Creation of effective software-oriented cryptalgorithms can promote solving numerous practical problems without utilizing any additional hardware resources, the following advantages being obtained:

- construction of crypto-protection mechanism for a wide use in different applications of information and telecommunication technologies;

- diminishing financial expenses on crypto-protection and on measures providing information safety;
- decreasing time requirements for development of protected information systems;
- possibility of development of a wide spectrum of official standards on cryptalgorithms to comply to special requirements of certain applications;
- easiness of standards modernization;
- high flexibility of protected systems for data processing and data transfer;
- convenient mechanism for automatic checking of reference state of technological programs and of source data;
- re-programmable ciphering devices could be created on the basis of serial microprocessors (in this case encryption standards can be modified without any re-organization of industrial basis).

Recently two fast software encryption functions have been proposed [4]. These algorithms having high encryption speed (4 - 8 Mb/s for SUN 4/260) are of great interest. One can mention some more cryptalgorithms suitable for soft implementation, for example RC5 [5].

On the basis of new fast software encryption systems with tuning subsystem [6, 7] a promising computer protection system COBRA [8] has been created. The high-speed cryptalgorithms used in this system have been studied in several special institutions and their high level of the cryptorobustness has been confirmed. These ciphers are based on utilization of the local pseudorandomness. Use of local uncertainty allows developing robustant cryptosystems with reduced complexity of the encryption procedures and enhanced rapidity.

This paper describes a new improved pseudoprobabilistic cipher and presents its general cryptorobustness characterization.

2 Block cipher with pseudoprobabilistic mechanism

In this cipher there are used unique combinations of the key area elements at every encryption step. The use of local uncertainty allows one to reduce at the average the number of conversion rounds providing high level of the cryptorobustness. The cryptoconversion procedure is divided into two stages: (1) initialization of the cipher, and (2) data encryption. During the first phase the password-controlled tuning of the resident cryptomodule is executed. The first phase subroutine is not critical regarding its size and processing time, because it is started only once, when entering the cryptosystem. On the contrary, the encryption proceedures must be realized economically and efficiently.

To avoid any detectable correlation between password and key area this algorithm is to be constructed in accordance with the following principles: (1) formation of the output parameters of the tuning subsystem must have pseudorandom character, (2) every bit of the password has to influence all bits of the key area of the cipher, (3) tuning algorithm must be composed in the form of a one-way function.

ALGORITHM 1: Key area construction

Two known standard pseudorandom sequences $\{Z_j\} = \{(z_1 z_2)_j\}$ and $\{Q_j\} = \{(q_1 q_2)_j\}, j = 0, 1, 2, \ldots, 511$, of 16-bit numbers are used.
INPUT: Password (source key) of arbitrary length.

1. The password is repeated several times to obtain the 1024-byte sequence $\{P_j\} = \{(p_1 p_2)_j\}$, where P_j is the 16-bit representation of the couples of corresponding characters.

2. Using the module 2 addition operation $\oplus$, an intermediate 1024-byte controlling sequence is formed $\{H_j\} = \{Z_j \oplus P_j\}$.

3. Set counter $i = 1$ and calculate initial values of the $(c_1 c_2)$, y and

u variables:

$$(c_1c_2)_{-1} = (p_1p_2)_{17},$$
$$y_0 = (p_1p_2)_{11},$$
$$u_0 = (p_1p_2)_{37}.$$

4. Calculate the current values of the Y and U variables:

$$Y_i = [y_{i-1} \oplus (p_1)_{i-1}] + F(u_{i-1}) \ mod \ 256^2,$$

$$U_i = [u_{i-1} + (p_2)_{i-1}] \oplus F(y_{i-1}) \ mod \ 256^2,$$

where $F(x) = H_x$.

5. Execute conversion

$$(c_1c_2)_{i-1} = \{[(p_1p_2)_{i-1} + Y_i] \oplus (c_1c_2)_{i-2} \oplus U_i\} \ mod \ 256^2$$

and calculate $u_i = U_i \ mod \ 512$, $y_i = Y_i \ mod \ 512$.

6. If $i < 512$ then increment i and jump to step 4.

7. Construct the second controlling sequence $\{D_j\} = \{Q_j + P_j \ mod \ 256^2\}$.

8. Set initial values $(k_1k_1)_{512} = (p_1p_2)_{209}$, $y_0 = (p_2)_{119}$ and $u_0 = (p_1)_{239}$.

9. Calculate the current values

$$Y_{i-1} = [y_{i-1} - (c_2)_{512-i}] \oplus F(u_{i-1}) \ mod \ 256^2,$$

$$U_{i-1} = [u_{i-1} \oplus (c_1)_{512-i}] - F(y_{i-1}) \ mod \ 256^2,$$

where $F(x) = D_x$.

10. Execute conversion

$$(k_1k_1)_{512-i} = \{[(c_1c_2)_{512-i} + (k_1k_2)_{513-i}] \oplus U_i\} + Y_i \ mod \ 256^2$$

and calculate $u_i = U_i \ mod \ 512$, $y_i = Y_i \ mod \ 512$.

11. If $i < 512$ then increment i and jump to step 9, otherwise STOP.

OUTPUT: 1024-byte key area $\{K_j\} = \{(k_1k_2)_j\}$, $j = 0, 1, 2, \ldots, 511$.

The output sequence of the Algorithm 1 is interpreted as a sequence of 32-bit words $\{K_h\} = \{(k_1k_2k_3k_4)_h\}$, $h = 0, 1, 2 \ldots 255$, and used as key area for the data encryption executed according to Algorithm 2.

ALGORITHM 2: Ciphering proceedures

INPUT: 512-byte data block $\{T_h\} = \{(t_1t_2t_3t_4)_h\}$, $h = 0, 1, 2, \ldots, 127$, where 32-bit words $\{T_h\}$ are represented as a concatenation of four bytes t_1, t_2, t_3, and t_4.

1. Define conversion mode: $E = 1$ (encryption) or $E = 0$ (decryption).

2. Set $r = 1$, $s = 4$ and define $\{R_h\} = \{T_h\}$.

3. Set counter $i = 1$, parameter $G_1 = K_{29}$, and initial values of the variables:

$$Y_0 = K_{93} + q \ mod \ 256^4,$$
$$U_0 = K_{29} \oplus 2^q \ mod \ 256^4,$$
$$n_0 = K_{15+q} \ mod \ 256, \ m_0 = K_{103+q} \ mod \ 256,$$

   where $F(x) = K_x$ and $q = 5(1 - E) + (2E - 1)r$.

4. Calculate the current variable values

$$n_i = (n_{i-1} \oplus i) + (u_2)_{i-1} \ mod \ 256,$$
$$Y_i = Y_{i-1} + F(n_i) \ mod \ 256^4,$$
$$m_i = [m_{i-1} + (u_1)_{i-1}] \oplus (y_2)_i \ mod \ 256,$$
$$U_i = [U_{i-1} + (G_i)^{<x<}] \oplus F(m_i) \ mod \ 256^4,$$

   where $x = 16(r - E)$.

5. If $E = 0$ then jump to step 8.

6. Calculate index $h = 32(r - 1) + (-1)^{r+1}i \ mod \ 128$.

7. Execute the current encryption step of round number r:

$$C_h = (R_h^{<x<} \oplus Y_i) + U_i \ mod\ 256^4,$$

where $x = U_i \ mod\ 2^5$, operator $< x <$ denotes to-left circular shift (x is the shift value in bits), and jump to step 10.

8. Calculate index $h = 32(4 - r) + (-1)^r i \ mod\ 128$.

9. Execute the current decryption step of round number r:

$$C_h = [(R_h - U_i) \oplus Y_i \ mod\ 256^4]^{>x>},$$

where $x = U_i \ mod\ 2^5$, $> x >$ denotes to-right circular shift operation.

10. If $i < 128$ then increment i, set $G_i = C_h$, and jump to step 4.

11. If $r < s$ then increment r, define new input data block $\{R_h\} = \{C_h\}$, $h = 0, 1, 2, \ldots, 127$ and jump to step 3, otherwise STOP.

OUTPUT: 512-byte data block $\{C_h\} = \{(c_1 c_2 c_3 c_4)_h\}$.

The program implementing the second algorithm provides the encryption speed about 20 Mbit/s for microprocessor Intel 486/100, with the size of its resident part being less than 2 kbytes. One can see that this algorithm uses a basic stream encryption procedure in four-pass mode (number of the passages is given by parameter $s = 4$).

3 Cryptanalytic estimations

One of the most complicated and most important problems for every new cryptosystem is the cryptoresistibility evaluation. As to the software-oriented cryptosystem described above at present no acceptable theoretical evidence of its high data protection level has been gained. On the other hand, investigation of different schemes for breaking this system has given no reasonable approaches to the latter problem. In this one the system under consideration resembles many other single key cryptalgorithms. Some attacks on the software-oriented cryptosystem are discussed below.

3.1 Password search attack

As compared with DES-like algorithms the two-stage cryptoscheme resists this attack much better. Indeed, the presence of the very prolonged tuning stage is the additional barrier counteracting this attack because to test a trial password one must run the tuning subroutine before executing check decryption.

Any attack using correlation between key area elements of the resident module and password seems to be unrealizable. It is prevented by the properly composed tuning subroutine algorithm which is a one-way function. In numerous experimental attempts we have not succeeded to reveal any correlation between key area and password. Any bit of the password influences all elements of the key area. Compression tests and statistical ones have shown that the tuning subroutine generates pseudorandom key sequences.

3.2 Statistic cryptanalysis of ciphertext

This method is hardly a reasonable way to compromise the system under discussion which transforms any type cleartext (including the repetition of the same symbol) into pseudorandom sequences of bytes. The frequency of all symbols of the ciphertext is about the same. A great many of cryptogramms were checked by several spectral tests which proved pseudorandomness of the ciphertext symbol distribution. Data compaction algorithms do not compress the cryptogramms.

3.3 Known plaintext attack

Every 32-bit word of the plaintext is converted in accordance with the following generalized formula

$$C_i = f(T_i, Y_{i1}, U_{i1}, Y_{i2}, U_{i2}, Y_{i3}, U_{i3}, Y_{i4}, U_{i4}), \qquad (1)$$

where $\{Y_{ik}, U_{ik}\}$, $k = 1, 2, 3, 4$, is a pseudorandom set of 32-bit values of the key variables Y and U. The Y_{ik} and U_{ik} values are defined by a combination of the key area elements K_j determined by corresponding input data block. Variation of any bit in the given input block results in

a pseudorandom change of these combinations. Though there are used only 256 of 32-bit key elements the data-dependent variables Y and U take on the pseudorandom values from the set $M = \{0, 1, 2, \ldots, (256^4 - 1)\}$, $\#M = 2^{32}$.

To calculate elements K_h of the key area it is necessary to determine the values Y_{ik} and U_{ik}. The lasts ones can be calculated from a system of the (1)-type equations. For this purpose one has to find the encryption steps corresponding to the same $\{Y_{ik}, U_{ik}\}$ sets. One must have two or more equations in such system the solution of which is not a complex problem from the cryptographical point of view. The main known plaintext cryptanalysis problem consists in detecting repetitions of the $\{Y_{ik}, U_{ik}\}$ sets.

Appearance of the given $\{Y_{ik}, U_{ik}\}$ set at the current encryption step has the probability $P = (\#M)^{-8}$. The minimal known plaintext size V_{min} which is necessary for cryptanalysis can be estimated by the formula corresponding to the 0.5 probability of the observation of a $\{Y_{ik}, U_{ik}\}$ set repetition:

$$V_{min} > V_{0.5} = \sqrt{P^{-1}} = (\#M)^4 = 2^{128}\ words \approx 10^{39} bytes. \qquad (2)$$

General approach to the repetition detection is to solve systems of several equations

$$\begin{cases} C_i = f(T_i) \\ C_j = f(T_j) \\ \cdots\cdots\cdots \\ C_g = f(T_g), \end{cases}$$

where $C_i \neq C_j \neq \ldots \neq C_g$, $T_i \neq T_j \neq \ldots \neq T_g$, $i, j, \ldots, g = 1, 2, 3, \ldots, N$, for every set of indexes $(i, j, \ldots, g)$. $N = (\#M)^4$ is the number of 32-bit words in the the known plaintext. In the most favourable case cryptanalyst can detect repetitions considering such system containing only two equations. Let us assume that he is able to reject successfully all false solutions. But to meet a true solution he has to check on the average more than $C_N^2/2$ of such systems. On the average the complexity of this procedure is

$$S_{min}^{(1)} = S_0 \cdot C_N^2/2 \approx (S_0/4) \cdot N^2 = (S_0/4) \cdot (\#M)^8, \qquad (3)$$

where S_0 is some average number of operations which are to be executed while checking one system. For $S_0 = 2$ we have $S^{(1)}_{min} > 10^{76}$ operations.

If a very large plaintext is known ($V \gg V_{min}$) then cryptanalyst can randomly select the $T_i \longrightarrow C_i$ pairs. In this case he has to try at the average $(\#M)^8 \cdot ln2$ variants per one $\{Y_{ik}, U_{ik}\}$ set repetition. For this case the minimal complexity is $S^{(*)}_{min} = S_0 \cdot (\#M)^8 \cdot ln2 > S^{(1)}_{min}$.

Analysis of the couples $T_i \longrightarrow C_i$ located at the positions from which the encryption of the 512-byte blocks begins gives some advantages since all words $\{T^{(w)}_0\}$, where w is the number of corresponding block, are encrypted with the same parameters Y_1 and U_1 in the first round. Searching of the $\{Y_{ik}, U_{ik}\}$ set repetitions for the sequence $\{T^{(w)}_0\}$, $w = 1, 2, 3, \ldots, W$ gives the following values $V^{(2)}_{min} = 128 \cdot (\#M)^3 words \approx 3 \cdot 10^{31} bytes$, $S^{(2)}_{min} > (S_0/4) \cdot (\#M)^6 \approx 10^{57}$.

3.4 Chosen text attacks

The chosen known plaintext and chosen cryptogram attacks are the most powerful cryptanalytic instruments. For these methods the most effective approach is to find a special input texts which will generate heavy the $\{Y_{ik}, U_{ik}\}$ set repetitions. For this purpose cryptanalyst can choose, for example, a set of plaintext (ciphertext) blocks $\{T^{(w)}_h\}$ ($h = 0, 1, 2, \ldots, 127$; $w = 1, 2, 3, \ldots$), where $T^{(w_1)}_h = T^{(w_2)}_h$ for $h \leq 30$ and $T^{(w)}_h$ are pseudorandom 32-bit words for $31 \leq h \leq 127$. In this case for all words $T^{(w)}_{31}$ ($w = 1, 2, 3, \ldots, W$) the parameters $Y^{(w)}_{31}$ and $U^{(w)}_{31}$ are fixed for the first and for the second encryption rounds. Hence for the sequence $\{T^{(w)}_{31}\}$ the appearence of the given set $\{Y_{ik}, U_{ik}\}$ has the probability $P = (\#M)^{-4}$. For $W_{min} = \sqrt{P^{-1}} = (\#M)^2 \approx 10^{19}$ the probability to find a $\{Y_{ik}, U_{ik}\}$ set repetition in the sequence $\{T^{(w)}_{31}\}$ equals 50%. The threshold value of the chosen text is $V^{(3)}_{min} = 128 \cdot W_{min} > 10^{21} words$. The minimal complexity of the one $\{Y_{ik}, U_{ik}\}$ set repetition detection is

$$S^{(3)}_{min} > S_0 \cdot (C^2_{W_{min}}/2) \approx (S_0/4) \cdot (\#M)^4 > 10^{37} operations. \qquad (4)$$

Determination of the key elements K_h from a set of supposedly found values Y and U is a new problem leading to increase of the values $V_{min}^{(3)}$ and $S_{min}^{(3)}$. To reduce local uncertainty one has to involve in simultaneous consideration several 32-bit known plaintext words as well as procedures generating the values of the key variables Y and U. This results in a drastic increase of corresponding equations complexity. It is not clear how one can reduce the local uncertainty to diminish the cryptanalysis complexity because such attempts results in very complicated encryption equations.

4 Conclusions

Use of pseudorandom selection of the key area element combinations gives the possibility to reduce the number of encryption proceedures and to increase conversion speed. New software encryption algorithm based on local uncertainty and resisting well different types of the cryptanalysis has been developed. In the proposed fast pseudoprobabilistic cryptosystem there are used two fixed operations (modulo-2^k addition and modulo 2 addition) and two data-dependent ones (circular shift and displacement selection of the key elements the last being a specific generalized operation). The robustness has been estimated from some general probabilistic point of view. The cipher seems to have high cryptorobustness level ($> 10^{57}$ operations for known plaintext attack and $> 10^{37}$ operations for chosen plaintext and chosen cryptogram attacks) the minimal size of the required text being extremely large ($> 3 \cdot 10^{31}$ bytes of the known plaintext or $> 4 \cdot 10^{21}$ bytes of the chosen texts). Software implementation of the proposed cipher provides about 20 Mbit/s encryption speed for Intel 486/100.

References

[1] Garon G., Outerbridge R. DES watch: an examination of the sufficiency of the Data Encryption Standard for financial institution in the 1990's // Cryptologia. 1991, Vol.15, No.3, p.177–193.

[2] Miyaguchi Sh. The FEAL cipher family // Lect. Notes Comput. Sci. 1991, Vol.537, p.627–638.

[3] Cusick T.W., Wood M.C. The REDOC II cryptosystem // Lect. Notes Comput. Sci. 1991, Vol.537, p.545–563.

[4] Merkle R. S. Fast software encryption functions // Lect. Notes Comput. Sci. 1991, Vol.537, p.476–501.

[5] Rivest R.L. The RC5 Encryption Algorithm // Dr.Dobb's Journal, Jan. 1995, p.146–148.

[6] Moldovyan A.A., Moldovyan N.A., Moldovyan P.A. A new method of cryptographical transformations for modern computer security systems//Upravlyayuschie sistemy i mashiny. 1992. No.9/10, p.44–50 (Russian).

[7] Moldovyan A.A., Moldovyan N.A. Fast Software Encryption Systems for Secure and Private Communication // 12th International Conference on Computer Communication "Information Highways for a Smaller World and Better Living", Seoul, Korea, 21-24 August 1995. Proceedings. p.415–420.

[8] Moldovyan A.A., Moldovyan N.A., Moldovyan P.A. Effective software-oriented cryptosystem in complex PC security software // Computer Science Journal of Moldova. 1994, Vol.2, No.3, p.269–282.

A.A.Moldovyan, N.A.Moldovyan,
Institute of Modelling and
Intellectualization of Complex Systems,
5, Prof. Popov str., St-Petersburg,
197376, Russia
phone: 7(812)2340415; fax: 7(812)2349093
e–mail: *sovetov@imics.spb.su*

Received 16 November, 1995