An Encryption Algorithm and Key-stream Generator for Chinese Text Messages by Character Internal Code Structure

Tak-Ming Law
Hong Kong Institute of Vocational Education (Morrison Hill)
Department of Computing, Email: tmlaw@vtc.edu.hk

Abstract. *This paper proposes an algorithm to encipher the Chinese plaintext message written in Big-5/GB Chinese character internal codes. Unlike the ordinary ciphers, the Crypto-key of our proposed algorithm consists of a pair of formulas and a set of parameter values. The senders can formulate and compose their own unique formulas and parameters for encryption. According to the character internal code structure, we apply the formulas in a Key-stream generator to encipher the Chinese plaintext message. Since the proposed stream generator does not contain* ***permanent*** *encryption and decryption operations, the opponents are inadequate to predict the forms of its output (ciphertext). Experiment results show that the proposed algorithm achieves the data secrecy.*

Key words: encryption, decryption, key-stream, Big-5/GB internal code, character, plaintext, ciphertext.

1: Introduction

As Internet technology widely adopted by societies, network security issue becomes essential in this century. Many people talk about data privacy and feeling it is necessary when they are sending or receiving information through Internets. Especially when E-commerce is getting more and more popular in marketing, data privacy is essential for both customers and wholesalers to protect their data against unauthorized people. Because of the increasing interest in the Internet, along with the extensive use of low-cost networking equipment for homes and small enterprises, E-commerce marketing grows rapidly. The explosive growth of networking technology creates the needs of the privacy and integrity of electronic data.

Fortunately, security algorithms do not have to be expensive or complicated. One of the alternatives to protect the privacy of information is to perform encryption process. Generally speaking, ciphers can be classified as two types: block and stream ciphers. Discussions are focused on the stream cipher in this paper. Some works of stream ciphers have been done, like self-shrinking generator [1], summation generator [2,3] and RC4 [4]. However, seldom of them mention about Chinese text data security. For this reason, we are proposing a Chinese text encryption algorithm and key-stream generator for those who want to encipher their Chinese messages before sending to the receivers. The senders just simply implement their own encryption engine at home according to the proposed algorithm; then the meanings of the message will be protected although it is wiretapped (attacked) by intruders in the network channels.

The remainder of this paper will be organized as follows: in the next section, the Chinese plaintext cryptographic algorithm will be introduced; in section 3, the Big-5/GB internal Code Encryption Algorithm is proposed, and finally some experimental results and discussions will be given in the section 4.

2: Chinese Plaintext Cryptographic Algorithm

The proposed algorithm is a stream cipher. Let M be the whole Chinese plaintext message. M is divided into successive partitions m_i, i.e., $m_1, m_2, \dots m_n$ and $1 \le i \le n$, where i is the i^{th} Chinese character and n is the total number of characters in M respectively and $m_1, m_2, \dots m_n \in M$. In other words, each m_i

stands for one Chinese character (16 bits) within the set of whole Chinese plaintext message M. The encryption engine enciphers each m_i (16 bits) at a time with the key-stream element k_j where $k_j \in K$, K is the entire key-stream and $1 \leq j \leq n$. The process of encryption is as follows:

Operation (1):

$$E_K(M): M \rightarrow C = e_{k1}(m_1)e_{k2}(m_2)........e_{kj}(m_n)$$

, where $E_K(M)$ and $e_{kj}(m_n)$ are the encryption of whole message and a single Chinese character plaintext respectively. For M and C are the whole Chinese message of plaintext and ciphertext respectively. As mentioned before, K is composed of j number of value k which is either ascending or descending and must not be equal to zero. We set j as 5 and k start from 1 ascending to 5 in our experiment which is:

Operation (2):

$$\overbrace{k_1 \cdots k_5}^{K} \overbrace{k_1 \cdots k_5}^{K} \overbrace{k_1 \cdots k_5}^{K} .... ,$$

i.e., K = (1,2,3,4,5).

Key-stream K is performed iterately by $\frac{n}{j}$ times and serving as a key-stream element k_j to each individual character encryption $e_{kj}(m_n)$ until the end of $E_K(M)$. However, if $j = n$ then the key-stream K will not be reoccurred during the encryption process.

The algorithm of deciphering a ciphertext $D_K(C)$ is the reversion of $E_K(M)$,that is:

Operation (3):

$$D_K(C): C \rightarrow M = d_{k1}(c_1)d_{k2}(c_2).....d_{kj}(c_n)$$

, where $D_K(C)$ and $d_{kj}(c_n)$ are the decryption of whole ciphertext and a single encrypted Chinese character respectively. Thus, operation **(3)** is the reversion of operation **(1)**.

3: Big-5/GB Internal Code Encryption Algorithm and the formation of Key-stream Generator

The structure Big-5/GB character internal code has two sections and each section carries a hexadecimal value. For examples, the Big-5 internal code **a5e0,** for the character 丞, can be divided into two sections **a5** (8 bits) and **e0** (8 bits) respectively. The structure looks like:

a5 (8 bit)	e0 (8 bit)

In the early stage of our experiments, we simply add k_j to the first section and subtract k_j from the second section, that is,

a5 + k_j	e0 - k_j

The addition and subtraction operations will iterate from $k_1 \cdots k_j$ and then K by K until the end of $E_K(M)$ (see operation **(2)**). For examples, assume we have 10 characters for encryption; j is set to 4 and the value of k is started from 1 ascending to 4. The operations will be like this:

$S1_1$ + 1	$S2_1$ - 1
$S1_2$ + 2	$S2_2$ - 2
$S1_3$ + 3	$S2_3$ - 3
$S1_4$ + 4	$S2_4$ - 4
$S1_5$ + 1	$S2_5$ - 1
$S1_6$ + 2	$S2_6$ - 2
$S1_7$ + 3	$S2_7$ - 3
$S1_8$ + 4	$S2_8$ - 4
$S1_9$ + 1	$S2_9$ - 1
$S1_{10}$ + 2	$S2_{10}$ - 2

, where $S1_i$ and $S2_i$ are the first and second sections of the Big-5/GB internal code of m_i respectively. The value of j and k can be adjusted any time in order to obtain different encryption effects.

In the later stage of our experiments, in order to extend the encryption output variations, we switch and manipulate the operators (+, -) with multiplication (*) and combine the operators in each section, like,

$((S1_i \;*\; k_j) + \alpha) - \beta$	$((S2_i \;*\; k_j) - \alpha) + \beta$

, where α and β are constants. We did not apply those operators, such as (/, exp, mod and div), which will cause data lose after recovery (deciphering). The manipulations of operators and constants combination (Formulas) in each section are unlimited. Therefore, the output variations (ciphertext format) of $E_K(M)$ are enormous. Basically speaking, the manipulations within each section of the character codes build up the formation of key-stream generator and the details will be discussed next.

4: Discussions and Experiment Results

We have chosen 5401 most usually used Chinese characters to test the algorithm. The 5401 characters were put into a text file and retrieved by the programs written in C language (see fig.1). The programs are able to encipher the original Chinese characters into a set of different characters (see fig. 2 and fig. 3). As we adjust the value of k , j, α, β, and manipulate the formulas within each section of the Big-5 internal codes, the unlimited output (ciphertext) variations had been generated. Therefore, it is unlikely for any intruders be able to decode the ciphertext.

4.1: Experiments for Key-stream Generator

The part of programs in Fig.1 was written according to the following formulas in the structure of Big-5 internal code (see the underline portion in Fig. 1):

$(S1_i \;+\; k_j) \;*\; \alpha$	$(S2_i \;-\; k_j) \;*\; \beta$

And the value of u, k , j, α, and β are written as e, kk, j, a and b in the program respectively and assigned by the parameters "1 1 5 7 9". Let u denotes an indicator that determines whether the value of k is ascending or descending; where value "1" and "0" of u indicate the ascending and descending property of k respectively. Then, the parameters "u k j α β" with the formulas in $S1_i$ (Section 1 of Big-5 internal code) and $S2_i$ (Section 2 of Big-5 internal code) become the key-stream K generator of both encryption and decryption of the whole Chinese plaintext message M . Thus, the key-stream K generator can be formulated in a single expression with seven parameters as:

$$\{u \;\; k \;\; j \;\; \alpha \beta \;\; FS1_i \;\; FS2_i\}$$

, where $FS1_i$ and $FS2_i$ are the Formulas in the first and second section of the Big-5/GB internal code of m_i respectively. The values of $\{u \;\; k \;\; j \;\; \alpha \;\; \beta \;\; FS1_i \;\; FS2_i\}$ should be random and must not repeat or be used to encipher another message M . Otherwise, the perfect secrecy, as Shannon [5] stated in the pioneer period of Cryptography, will not be achieved. In order to achieve maximum security of data, the key stream should contain no period (which is repeated identical encryption within a single key stream).

```
:
main(argc,argv)
/* Get the Parameters from the users */
int argc;
char *argv[];
{
  int total=0, e=1 ;
  if (argc != 7)
  {printf("Example usage: c>en 1 1 5 7 9 file.txt");
  /* Make sure the argument input is correct */
   exit();}
  kk=atoi(argv[2]); j=atoi(argv[3]);
  a=atoi(argv[4]); b=atoi(argv[5]);
  e=atoi(argv[1]); k=kk;
/* Assign the arguments value to the appropriate
variables */
  printf("%x%x=%c%c ", first,second,
      (((first+k) * b)  ),((second-k)) * a);
   if  (e == 1)
        {if (k>kk+j) k=kk; k++;}
        /*If ascending is indicated */
   else
         { if (k < kk-j) k=kk; k--;}
         /* If descending is indicated */
:
```

Fig. 1 The part of program that utilizing the key-stream K generator to encipher the Chinese Characters.

a440=一	**a441=**乙	**a442=**丁	**a443=**七	**a444=**乃
a445=九	**a446=**了	**a447=**二	**a448=**人	**a449=**儿
a44a=入	**a44b=**八	**a44c=**几	**a44d=**刀	**a44e=**刁
a44f=力	**a450=**匕	**a451=**十	**a452=**卜	**a453=**又
a454=三	**a455=**下	**a456=**丈	**a457=**上	**a458=**丫
a459=丸	**a45a=**凡	**a45b=**久	**a45c=**么	**a45d=**也
a45e=乞	**a45f=**于	**a460=**亡	**a461=**兀	**a462=**刃
a463=勺	**a464=**千	**a465=**叉	**a466=**口	**a467=**土
a468=士	**a469=**夕	**a46a=**大	**a46b=**女	**a46c=**子
a46d=孑	**a46e=**孓	**a46f=**寸	**a470=**小	**a471=**尢
a472=尸	**a473=**山	**a474=**川	**a475=**工	

Fig. 2 Original Chinese characters corresponding to Big-5 internal codes.

a440=苧	**a441=**砦	**a442=**栯	**a443=**頫	**a444=**髀
a445=	**a446=**箪	**a447=**痄	**a448=**魱	**a449=**鶷
a44a=	**a44b=*ÿ**	**a44c=*ÿ**	**a44d=*ÿ**	**a44e=*ÿ**
a44f=*ÿ	**a450=*"**	**a451=*"**	**a452=*"**	**a453=*"**
a454=*"	**a455=**淍	**a456=**竫	**a457=**踗	**a458=**塹
a459=	**a45a=**狷	**a45b=**罫	**a45c=**鋏	**a45d=**錍
a45e=	**a45f=?**	**a460=?**	**a461=?**	**a462=?**
a463=?	**a464=**眳	**a465=**薔	**a466=**霈	**a467=**餺
a468=	**a469=**笤	**a46a=**蔆	**a46b=**鴑	**a46c=**鶝
a46d=	**a46e=**羝	**a46f=**裎	**a470=**儜	**a471=**嚁
a472=	**a473=***□	**a474=***□	**a475=***□	

Fig. 3 Enciphered Chinese characters corresponding to Big-5 internal codes utilizing j which is set to 5, the value of k is started from 1 ascending to 5, the value of α and β are 7 and 9 respectively, by using the key-stream K generator $\{u\ k\ j\ \alpha\ \beta\ FS1_i\ FS2_i\}$, i.e. the parameters of "1 1 5 7 9".

The above test is only served as an example. Actually, senders can design better formula structures in different ways by increasing the number of parameter constants. Moreover, bit-wise exclusive-OR and bit rotation (left or right) can also be used in the formulas, like:

$((S1_i + k_j)*\alpha) \oplus k_j$	$((S2_i - k_j)*\beta) \oplus k_j$

, where $\oplus$ is denoted as exclusive-OR. One can also use bit rotation (left or right) to design the formulas in the same manner.

4.2: Conclusions

The requirements of the proposed algorithm are (1) both the senders and receivers need to have the identical functioning programs for enciphering and deciphering, (2) the receivers must have obtained the key-stream K generator (i.e., $\{u\ k\ j\ \alpha\ \beta\ FS1_i\ FS2_i\}$) before the receipt of ciphertext from the senders. Therefore, a good key exchange protocol, like Diffie-Hellmen scheme [6], must be performed. We shall have further development on this issue, (3) the algorithm only enciphers the Chinese text in M and ignores the other characters, e.g. punctuation and alphabets.

Although the algorithm is hardly perfect, however, it provides an easy way to achieve sufficient protection for ordinary emails or messages. The flexibility of senders to design their own formulas along with parameters within the encryption key becomes the significance of this paper. This algorithm also simplifies the dimensions of encryption for the data used in E-commerce. Although it still remains a lot of problems to be solved, the current results are encouraging and inspire us to put further effort to discover more solutions.

Reference

[1] W. Meier and O.Staffelbach., "The self-shrinking generator," In Advances in Cryptology---Eurocrypt '94, Springer-Verlag.

[2] O. Staffelbach and W. Meier. "Cryptographic significance of the carry for ciphers based on integer addition," In A.J. Menezes and S.A. Vanstone, editors, Advances in Cryptology --- Crypto '90, pages 601-615, Springer-Verlag, New York, 1990.

[3] W. Meier and O. Staffelbach. "Correlation properties of Combiners with memory in stream ciphers," In I.B. Damgard, editor, Advances in Cryptology --- Eurocrypto '90, pages 549-562, Springer-Verlag, Berlin, 1991.

[4] R.L. Rivest. "The RC4 Encryption Algorithm," RSA Data Security, Inc., March 12, 1992.

[5] C.E. Shannon, "Communication Theory of Secrecy Systems, " Bell Syst. Tech.J. Vol.28 pp. 656-715 (Oct. 1949).

[6] W. Diffie and M. Hellman, "New Directions in Cryptography,", IEEE Trans. on Info. Theory Vol. IT-22(6) pp. 644-654 (Nov. 1976).