ENCRYPTION TECHNIQUES IN DATA COMMUNICATION NETWORKS

BY

OKAFOR, ERIC C.
Department of Computer Science And Engineering
Enugu State University of Science and Technology, Enugu
e-mail – cue@infoweb.abs.net, ericcokafor@yahoo.com

ABSTRACT

Electronic messaging and data transfers are the main features of most communication networks. Electronic messaging (e-Mail) in a network environment for example has grown, providing features from ***simple correspondence, broadcasting (message distribution) and scheduling/calendering to on-line conferences*** with network users. Because these features are provided through a ***global data communication highway,*** confidentiality and data security is not always assured.

In most communication systems, the solution uses different methods to **scramble** the data to be transmitted thus providing secrecy to every correspondence.

This paper presents some techniques suitable for data encryption in any computer network. The presentation is very detailed and the techniques are described further using procedures written in a structured language.

INTRODUCTION

Encryption of messages is not new to digital systems. Mails and other correspondences exchanged through computer networks must be as confidential as if post or hand delivered them.

In most computing environments, the first level of access restriction uses *passwords*. Incorrect password is generally recognized by a program and used to lockout unauthorized users, a fairly effective way of stopping trespassers.

This form of security however may not be safe enough for a network environment where systems share information through a global information highway.

A network environment thus requires *top-notch security,* which demands programs that can *encrypt* entire messages during transmission and in storage using any reliable technique. Such programs are generally written as subroutines, and invoked automatically or selectively for data communication and storage. The decipher routine at the other end performing a reverse function recovers the message and could provide added security if program control allows it to be invoked automatically by a password.

This arrangement ensures that any transmission intercepted by spying users on the highway reserves its confidentiality.

ENCRYPTION ALGORITHMS

Two fairly straightforward algorithms will be presented. They are the *search-and-replace* and *one-time pad* techniques.

SEARCH-AND-REPLACE TECHNIQUE

This follows closely the search-and-replace function found in most word processors. It is illustrated in Figure 1. A block of one-digit random numbers is formed with certain numbers (selected keys) stripped off the block to produce a grille. This forms the *Cipher.* The message is inserted horizontally with each character occupying the position of a number. The transmitted **text** is then drawn from the block by accessing the grille vertically. Transposing the first and last characters with the adjacent characters completes *enciphering.*

At the destination, the original text is recovered by simply reversing this sequence. However, source and destination must operate on the same random number block RND(M, N) to ensure the deciphering of received data.

This technique is quite safe provided RND(M, N) remains known to only source and destination. However, in a situation where this leaks to other users, a change is quikly made to ensure data security. It is also possible to vary the data insertion and accessing technique or to use different random-number blocks within a particular text as a further security measure. A procedure already tested by the author automatically switches from one random block to another after enciphering a certain number of characters.

The procedure presented in Figure 2 implements the search-and-replace algorithm.

PROCEDURE FOR SEARCH-AND-REPLACE

In the procedure, a set of random numbers from 0 to 9 is used to define a two-dimensional array, RND (M, N). In the first loop of the procedure, this is translated to a grille that retains only the selected keys (in this example 1, 3, 8, 9).

The text is then processed as a string (list) of X characters. The second loop does this by replacing every element in the grille that is not 'space' by the next character in the text. Note that the processing is horizontal. The third loop in the procedure then assembles the elements in the array into a list DATA(y) and subsequently transposes the first and last characters with the adjacent characters. This enciphered list DATA(y) is then transmitted.

This method has the advantage of *ease of implementation* and will suffice for most applications demanding a fairly high level of security. The problem with this technique is that it transmits (or stores) the actual characters forming the original text.

ONE-TIME PAD TECHNIQUE

The one-time pad technique is more secure than the previous option. It uses random numbers on a pad for encryption. The text is first converted to numbers (example ASCII) as shown in Figure 3. Random numbers are then generated and added to this sequence. The sums are transmitted as the cipher text. Like in the search-and-replace technique, the pad of random numbers must also be communicated to the receiving system. Provided this pad is kept secret, this method is almost full proof.

PROCEDURES FOR ONE-TIME PAD

Figure 4 shows the *encipher* and *decipher* procedures for this method. Note that the *decipher* procedure reverses the operation at the enciphering end, so only its description is presented. To decipher the text, the transmitted information is first accumulated and the pad of random numbers recovered from storage.

A loop then inputs each transmitted code CIPHER(y) and converts this to the original data by subtracting the corresponding random number RND(y). The resulting codes (ASCII say) are then converted to the character equivalent and printed. This repetitive process terminates at the end of the transmitted sequence.

Complex variations of the above technique can be used to provide higher levels of security. One option changes the random numbers used after enciphering a specific number of characters within the text.

Also, instead of just adding the text code to the corresponding random number, a variation may add these two and subtract another random number or a value derived by applying a certain function on the text code.

Since the transmitted cipher text is the result of applying a certain function on the original text, recovery of the intelligence without prior knowledge of this function becomes impossible.

"ILLUSTRATING ENCRYPTION" ← *TEXT OR MESSAGE*

18439	→	18 39	→	I L L U
32931		3 931		S T R A
76025				
87069	*REMOVE*	8 8 9	*PROCESS*	T I N
13039	*KEY*	13 39	*MESSAGE INTO*	G E N C
92884	*NOS*	9 88	*GRILLE*	R Y P
01215	*1, 3, 8 ,9*	1 1	*HORIZONTALLY*	T I
95243		9 3		O N

Random no. block

Assemble Text Vertically → **ISTGROLETTIYLRNPIUANCN**

Transposing first and last characters with neighbours → SITGROLETTIYLRNPIUANNC
enciphered text

Figure 1. Encryption of message using Search-and-Replace method.

Pseudo code:

```
proc encipher
    define array rnd(m, n): [array of one-digit random numbers]
    [retain selected key numbers say 1,3,8,9 to produce new grille]
    col = 1
    while col <= n do
        row = 1
        while row <= M do
            if rnd(row, col) <> 1 or 3 or 8 or 9 then
                rnd(row, col) = space
            row = row + 1
```

```
            wend
            col = col + 1
        wend

[text is processed as a string (list) of x characters]
        get text (x)
        row = 1 : y = 1
        while row <= n do
        col = 1
            while col <= m do
                if rnd(row, col) <> space then
                    rnd(row, col) = text (y)
                    y = y + 1
                    col = col + 1
            wend
            row = row + 1
        wend
[transpose first and last letters with neighbours]
        define transmit data buffer data (x)
        col = 1 : y = 1
        repeat
            row = 1
                repeat
                    if rnd(row, col) <> space then
                        data(y) = rnd(row, col)
                        y = y + 1
                    row = row + 1
                until y > x
                col = col + 1
        until y > x
        swap data (1), data (2)
        swap data (x), data (y)
```

```
[transmit message]
        y = 1
        while not end_of_file
                xmit data (y)
                y = y + 1
        wend
corp
```

Figure 2. Procedure for Enciphering by Search-and-Replace Method.

"EXAMPLE OF ENCRYPTION" ← ***MESSAGE OR TEXT***

E	**X**	**A**	**M**	**P**	**L**	**E**		**O**	**F**		
69	88	65	77	80	76	69	32	79	70	32	*TEXT NOS*
06	09	20	17	17	16	05	13	00	15	27	*RANDOM NOS*
75	***97***	***85***	***94***	***97***	***92***	***74***	***45***	***79***	***85***	***59***	*CIPHER TEXT*

E	**N**	**C**	**R**	**Y**	**P**	**T**	**I**	**O**	**N**	
69	78	67	82	89	80	84	73	79	78	*TEXT NOS*
01	07	04	13	01	19	06	22	11	12	*RANDOM NOS*
70	***85***	***71***	***95***	***90***	***99***	***90***	***95***	***90***	***90***	*CIPHER TEXT*

TRANSMITTED CIPHER TEXT

→ ***759785949792744579855970857195909990959090***

Figure 3. Encryption of Message using one-time Pad Method

```
[encipher procedure]
proc encipher
 [text (x) is the message file with a string of a characters]
 [rnd (x) is a list of x two-digit random numbers]
```

```
[standard ASCII table is used for converting text to numbers]
    define rnd(x)
    get text(x) : y = 1
    while not end_of_file do
        data(y) = asc(text(y))
        cipher(y) = data(y) + rnd(y)
        xmit cipher(y)
        y = y + 1
    wend
corp
```

(a)

```
[decipher procedure]
proc decipher
 [accumulate transmitted message]
    define rnd(x)
    y = 1
    while not end_of_file do
        input cipher(y)
        data(y) = cipher(y) - rnd(y)
        text(y) = chr$(data(y))
        print text(y)
        y = y + 1
    wend
corp
```

(b)

Figure 4. Procedure for One-Time Pad Method, (a) encipher and (b) decipher.

CONCLUSION

The need to remove *vulnerability and maintain data security* is premium in any data communication network. The use of computer networks for electronic messaging (e-Mail) has made it possible for users to exchange simple messages without having to send them by hand or through messengers. The radius here is global, covering both Intranet and Internet communications.

This paper has presented *cryptographic systems adaptable to any of digital communication network.* Routines based primarily on these algorithms have been designed, tested and are already in use.