An Algorithm To Reduce The Size Of Cipher Text

[1]Mani Arora , Dr. Derick Angles[2]

GJCST Computing Classification
E.3 , E.4

Abstract- **Globalization has increased the degree of connectivity of communication networks. Organizations are spending a huge amount for securing their data and information. The rising competition in the global market has increased the threats to access the sensitive data. Whether data is at rest or mobile data, it is the need of the hour to secure it from unauthorized access. Over the time a number of techniques and algorithms have been proposed for its security. A technique, which is good for one type of application, may not be effective and efficient for other applications. However, all the existing techniques suffer from one common drawback that is the size of the Cipher text, which is always same or more than the size of the Plain text. We in this research paper, qualitatively, emphasize the need of securing the data but keeping the size of Cipher text under check. We propose a new technique and algorithm to encrypt the Plaintext into a reduced size Cipher text.**

Keywords- Algorithm, reduce, cipher, size, text, security.

I. Introduction

Globalization has increased the degree of connectivity of communication networks. Organizations are spending a huge amount for securing their data and information. The rising competition in the global market has increased the threats to access the sensitive data. Unauthorisedly. People are busy in developing [1] new and advance Cryptographic techniques to secure their data. The Cryptography has a history over 4000 years. A single security technique cannot be the best for all the applications. So far in the history of Cryptography only one point is kept in mind while developing a Cryptographic protocol that is —to provide adequate security —The purpose of our paper is to point out the need of modern time Cryptography. This paper emphasizes the requirement of new techniques of Cryptography, which with the basic goal of providing adequate security must also be efficient in reducing the size of Encrypted Text referred as Cipher Text. We have used the term —reduced size cipher text —to highlight the main aim of this proposed technique.In order to develop such a technique one must take care of the fact that Cryptography is an effective and the only practical way to ensure secure communication over open communication networks. While keeping in check the size of cipher text we can't compromise with security of data and information. Further there must be no loss of text in the whole process of conversion of plaintext into reduced size cipher text.There is a reason for this and that is the fact that we can't afford to miss even a single number or character value while dealing with the text.

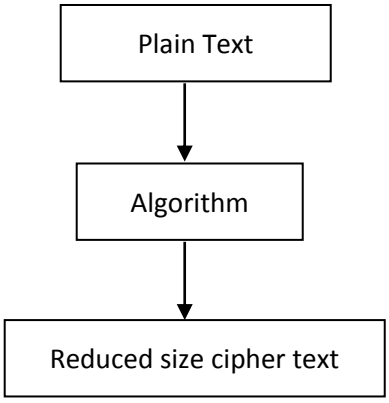
[1]Mani Arora Assistant Professor ,Khalsa College Amritsar (Mobile:9814088687,mani_mcain@yahoo.com)
[2]Dr.Derick Angles Professor,Guru Nanak Dev University (Mobile :9815952913,derickengles@yahoo.com)

II. Need of Proposed Technique

In general every cryptographic algorithm produces a cipher text, which has got the size equal to or more than the size of original plaintext. At the same time, it must be noted that when the cipher is decrypted then we will get the same plaintext, as it was original. But there is need for us to focus at cipher text. We can use it for when, we are sending or receiving it for a secured communication with other parties. For fast communication the use of reduced size cipher with the same message or information with the original text is decisively preferable. Moreover it will help in reducing the communication cost involved as well as of storage. When we are dealing with text each & every bit counted carefully, the loss of even a single bit can change the meaning of the text subsequently causing it misleading.

III. Introduction to Proposed Technique

The proposed technique intends to encrypt the plain text with a prechosen mathematical function along with the objective of reduced size cipher text.



The proposed algorithm works as in figure. In the technique we have maintained two dictionaries.

A. Primary Dictionary

The dictionary contains words, which are probably most frequently used along with the codes*(which will be explained later). The dictionary will be fixed in size & the codes too. Even if someone cracks the dictionary and codes, still our technique is going to work because it is based on both primary and secondary dictionary and secondary dictionary is not fixed.To maintain this dictionary plain text file is used and for processing we read the file into memory as an associative arrays. Associative arrays maps arbitrarily typed objects to arbitrarily typed objects. Data structures used to represent associative array when initialized in

memory will be linked list. For searching a word in the dictionary, simple linear search is used.

B. *Secondary Dictionary*

The dictionary is not fixed. This dictionary will be empty when initialized. Every time the algorithm come across the string in pass 2,it will add it to dictionary and assign a code to it. Starting code for first string in secondary dictionary will be fixed. The next codes can be obtained by doing increment of one step. As this dictionary will be created during runtime so it's difficult to crack the encryption. This dictionary will also be stored in same file that for primary dictionary is used.

IV. ENCRYPTION ALGORITHM

The algorithm will start with initializing a primary dictionary and variable S, which will be initially empty. We read the file containing dictionary into memory in linked list structure. The variable S will read plain text data file word by word and comparisons in dictionary will be made with help of this variable.

In the algorithm each word of plain text will be first searched in primary dictionary, this is done using linear search. If the word is present in primary dictionary then it will be replaced by corresponding code assigned to it and stored in encoded output file. Codes are in binary as binary take less memory space than any other data type. Codes are fixed for each data word so it can't be changed later on. Size of code allocated to each string will be 12 bits that is any dictionary can contain maximum 4096 entries

If the word is not found in primary dictionary then it is searched in secondary dictionary. If the word is present in secondary dictionary its corresponding code will be substituted to encoded output file, if not present then that word is substituted in secondary dictionary and the new code is generated to it by incrementing the code by 1 of last entry in secondary dictionary. The code for first entry in secondary dictionary will be fixed and rest codes will be obtained by incrementing each code by 1.Secondary dictionary will be of variable size.

After these pass we will get encoded output file. Further in the algorithm we use a predetermined mathematical function XOR to further encrypt the encoded output. As code assigned to each word is of 12 bits, so in pass3 each 12 bit block will be XOR with a secret key to get final output which will be a cipher text in reduced size then plain text.

So In Whole The Algorithm For Proposed Technique Is

i. Initialize S as an empty string
ii. Read primary dictionary in memory
iii. Do till EOF Read the next word from the file If this word is in primary dictionary get corresponding code from primary dictionary.Write the code in the output file else Read secondary dictionary in file

If this word is in secondary dictionary see the corresponding code from secondary dictionary Write the code in the output file else add this word to secondary dictionary

assign the next code obtained by incrementing previous code by 1 used to substitute the new code in the output file

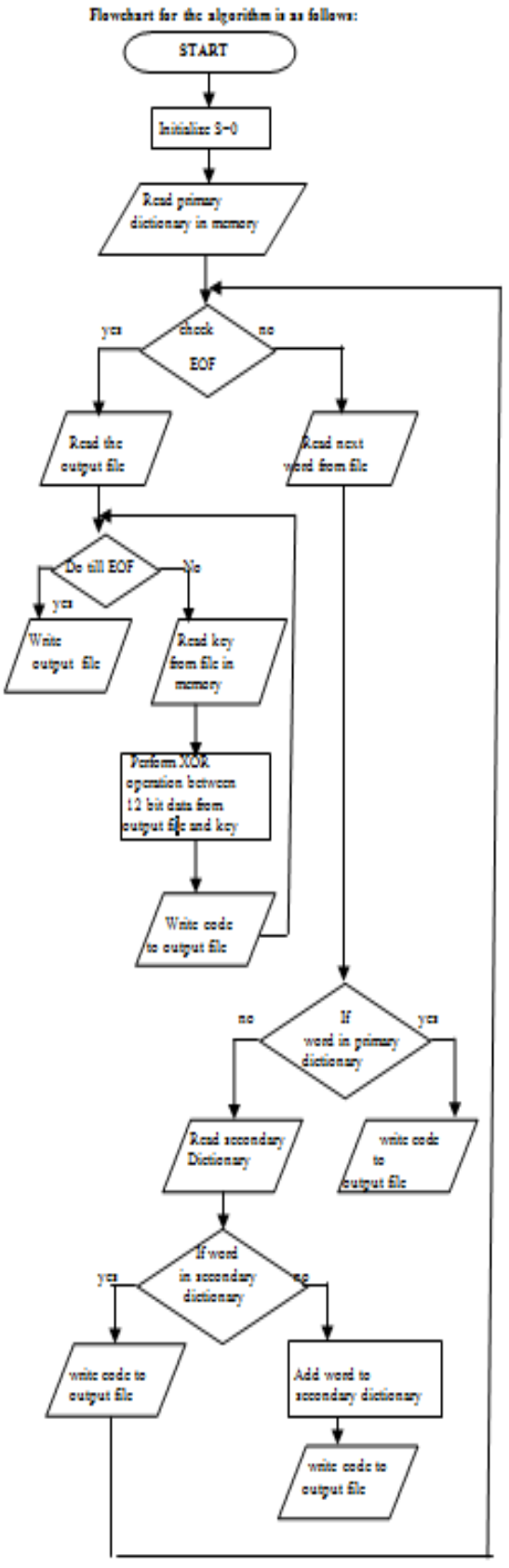
iv. Read the output file
v. Do till EOF Read key from file in memory Read next 12 -bit block from file Perform XOR operation between key and 12-bit block Write the code in final output file

Algorithm for Primary dictionary

Primary dictionary is fixed so sender and receiver both before transmission of cipher text will know it.

Algorithm to create primary dictionary

i. Start with prechosen code and assign first word to it.
ii. Add new word to dictionary.
iii. Assign code to new word by incrementing previous code by 1



Example-

Example of this technique is given below.

Plain text that is to be encrypted is-

The information cannot be understood by anyone for whom it was unintended. Confidentially is the protection of transmitted data from passive attacks. The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source & destination frequency length or other characteristics of the traffic on a communication facility:

Pass1

000000010001	000100000001
000000110011	000000001011
000100000010	000000001010
000100000011	000000010100
000000101110	000000001100
000000010010	000100000100
000000111001	000100000101
000000000100	000000010001
000100000110	000000001000
000100000111	000100001000
000000100010	000100001001
000100001010	000000111001
000000010001	000000000100
000100001011	000000001000
000100000101	000000000100
000000010001	000100000110
000000001000	000100001100
000000100010	000100001101
000000111001	000000100001
000100001110	000000100000
000000000011	000100001111
000000010101	000000001011
000000101001	000000001101
000100010000	000000010001
000100010001	000000001110
000100010010	000100010011
000000001111	000100010100
000000000111	000000101111
000100010101	000000001000
000000010001	000100010110
000000001001	000000000001
000100010111	000100011000

Pass2 In this pass each 12-bit block will be XOR with key, which is 111100001111

111100011110	111000001110
111100111100	111100001100
111000001101	111100000101
111000001100	111100011011
111100000001	111100000011
111100011101	111000001011
111100110110	111000001010
111100001011	111100011110
111000001001	111100000111
111000001000	111000000111
111100101101	111000000110
111000000101	111100110110
111100011110	111100001011
111000000100	111100000111
111000001010	111100001011
111100011110	111000001001

111100000111	111000000011
111100101101	111000000010
111100110110	111100101110
111000000001	111100101111
111100001100	111000000000
111100011010	111100000100
111100100110	111100000010
111000011111	111100011110
111000011110	111100000001
111000011101	111000011100
111100000000	111000011011
111100001000	111100100000
111000011010	111100000111
111100011110	111000011001
111100000110	111100001110
111000011000	111000010111

Contents Of Key File

Primary Dictionary

Code	String
0000 0000 0001	a
0000 0000 0010	as
0000 0000 0011	an
0000 0000 0100	is
0000 0000 0101	in
0000 0000 0110	if
0000 0000 0111	or
0000 0000 1000	of
0000 0000 1001	on
0000 0000 1010	by
0000 0000 1011	be
0000 0000 1100	it
0000 0000 1101	to
0000 0000 1110	&
0000 0000 1111	,
0000 0001 0000	can
0000 0001 0001	the
0000 0001 0010	was
0000 0001 0011	are
0000 0001 0100	for
0000 0001 0101	not
0000 0001 0110	and
0000 0001 0111	has
0000 0001 1000	one
0000 0001 1001	two
0000 0001 1010	too
0000 0001 1011	its
0000 0001 1100	who
0000 0001 1101	his
0000 0001 1110	her
0000 0001 1111	were
0000 0001 0000	that
0000 0010 0001	this
0000 0001 0010	from
0000 0001 0011	into
0000 0001 0100	with
0000 0001 0101	most
0000 0001 0110	here
0000 0001 0111	back
0000 0001 1000	form
0000 0001 1001	able
0000 0001 1010	must
0000 0001 1011	where
0000 0001 1100	which
0000 0001 1101	there
0000 0001 1110	whom
0000 0001 1111	other
0000 0011 0000	did
0000 0011 0001	should
0000 0011 0010	could
0000 0011 0011	cannot
0000 0011 0100	does
0000 0011 0101	he
0000 0011 0110	she
0000 0011 0111	—
0000 0011 1000	‘‘
0000 0011 1001	.
0000 0011 1010	always
0000 0011 1011	almost
0000 0011 1100	also
0000 0011 1101	national
0000 0011 1110	International
0000 0011 1111	like

Secondary Dictionary

0001 0000 0001	Information
0001 0000 0010	understood
0001 0000 0011	anyone
0001 0000 0100	unintended
0001 0000 0101	confidentially
0001 0000 0110	protection
0001 0000 0111	transmitted
0001 0000 1000	data
0001 0000 1001	passive
0001 0000 1010	attacks
0001 0000 1011	aspects
0001 0000 1100	flow
0001 0000 1101	analysis
0001 0000 1110	requires
0001 0000 1111	attacker
0001 0001 0000	observe
0001 0001 0001	source
0001 0001 0010	destination
0001 0001 0011	frequency
0001 0001 0100	length
0001 0001 0101	characteristics
0001 0001 0110	traffic
0001 0001 0111	communication
0001 0001 1000	facility
1111 0000 1111	Key

V. Conclusion

In this paper we have presented a completely new encryption algorithm that focuses on reducing the size of cipher text. Such technique will be helpful in saving the communication cost as well as storage space required in computer.

VI. REFERENCES

1) Mc Conell, Mike. & Harmittion, Booz Allen . (2002) —Information Assurance in twenty-first century — IEEE Security and Privacy, pp. 16 -18.
2) Adams, C.M.& Tavares, S.E. (1993) —Designing S-Boxes for Ciphers Resistant to Differential Cryptanalysis —proceedings of 3rd Symposium on state and Progress of Research in Cryptography, pp.181-190.
3) D.K. Branstad, D.K., Gait, J., and Katzke, S. (1976) —Report on the Workshop in support of computer security" in National Bureau of standards, p77-91.
4) Rijndael (1998) "Rijndael AES proposal" National institute of science and technology. Available [online] http://www. Csrc.nist.gov/encryption/aes/.
5) 5.Rivest, R.L. , Shamir, A. & Adlenman, L.M. (1978) —A method for obtaining digital signatures and Public-Key Cryptosystem —, Communications of the ACM, Vol.21, No.2, pp.120 –126 .
6) Schneier, B. (2001) –Applied Cryptography", (John Wiley and Sons) p11-13.