ISSN:0975-9646

ENCIPHER A Text Encryption and Decryption Technique Using Substitution-Transposition and Basic Arithmetic and Logic Operation

Devendra Prasad#1 ,Govind Prasad Arya#2, Chirag Chaudhary#3, Vipin Kumar#4

#1234 Department of Computer Science & Engg.,Uttarakhand Technical University
Dehradun, Uttarakhand, India

***Abstract*— In this age of universal electronics connectivity, of viruses and hackers, of electronic eavesdropping and electronic fraud, there is indeed no time at which security does not matter. The explosive growth in computer systems and their interconnections via network has increased the dependence of both organizations and individuals on the information stored and communicated using these systems. This, in turn, has led to a heightened awareness of the need to protect data and resources from disclosure, to guarantee the authenticity of data and messages, and to protect systems from network-based attacks. The disciplines of cryptography and network security have matured, leading to the development of practical, readily available applications to enforce network security. In this paper we have proposed a cipher which uses basic encryption techniques of substitution and transposition along with application of logic gates, in order to encrypt the data. The algorithm makes cryptanalysis even more difficult because of the use of "Random Number Generator" function which further decides order of encryption rounds and keys to be used to encrypt the plain text. This eliminates the overhead of defining a fixed key by the user and makes algorithm secure also. It also facilitates to transfer the key to the receiver while being added with the plain text at random locations (like added at end or beginning).**

***Keywords*— Cipher, Cipher text, Decryption, Encryption, Information Security, Key, Plaintext, Random number, Substitution, Transposition, Logical NOT operation, modulus function.**

I. INTRODUCTION

The requirement of **information security** within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. An example of the former is the use of rugged filing cabinets with a combination lock for storing sensitive documents. An example of letter is personnel screening procedures used during the hiring process.

With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone network , or the Internet. The generic name for the collection of tools designed to protect data and to thwart hackers is **computer security**.

The second major change that affected security is the information of distributed systems and the use of networks and communications facilities for carrying data between terminal user and computer and between computer and computer. Network security measures are needed to protect data during their transmission. In fact, the term **network security** is somewhat misleading, because virtually all business, government, and academic organizations interconnected their data processing equipment with a collection of interconnected networks. Such a collection is often referred to as an internet, and the term **internet security** is used.

There are no clear boundaries between these two forms of security. For example, one of the most publized types of attack on information systems is the computer virus. A virus may be introduced into a system physically when it arrives on a diskette or optical disk and is subsequently loaded onto a computer. Viruses may also arrive over an internet. In either case, once the viruses is resident on a computer system, internal computer security tools are needed to detect and recover from the virus.

II. SHORTCOMINGS OF PREVIOUS ALGORITHM

- The previous algorithm makes use of a fixed key initially. This fixed key is defined by the user itself which can become overhead for the user.
- The algorithm generates five keys for the five different rounds, which are the first to fifth multiple of the initial key. Therefore if the initial key is known by attacker, all the other five keys can be known.
- Since the orders in which rounds are applied in algorithm are always fixed, hence decryption becomes easy.
- The algorithm can only encrypt and decrypt Capital letters which can easily intruded.
- The previous algorithm uses user defined fixed letters range (for e.g. A=1, B=2 ….Z=26.)
- The time complexity of the previous algorithm was affected by this user defined letter range.

III. Our Contribution

An The algorithm proposed by us uses the keys for encryption, which are generated from the message itself and are not required to be defined by the user whereas in the previous algorithm the initial key was supposed to be defined by the user explicitly. Once the encryption is done, the key is to be transferred to the receiver's end so that it could be used for decryption. Therefore it is transferred to the receiver's end while being added with the message in the encrypted form. Another role is played by random number generator to enhance security. The algorithm uses the substitution and rail-fence technique but the random number decides that which one of the two encryption techniques has to be applied first. The length of the original message decides the key to be used for substitution encryption. After this when both the algorithms have been applied, we apply NOT gate to each character. If the length of message is even, the key will be added at the end and the notation used for random number will be placed at the beginning of message in a byte else the notation will be stored at the end and key at the beginning. The notation for random number will be zero if it is even and one if it is odd. The key will also be stored in a byte using the five LSB of the word. The final message will be transmitted over the network.

The decryption algorithm on the other end will separate the key and notation used for random number from the cipher text by counting its length. Once they are separated, the cipher text will undergo NOT operation and decryption rounds will be applied subsequently, on the basis of random number notation. If random number notation is zero, rail-fence will be applied first and then substitution, else vice-versa.

IV. Encryption Algorithm

Step 1: Generate a Random Number R.
Step 2: If R is Even, go to **Step** 3.
Else: go to **Step** 9.
Step 3: Count the length of String
Step 4: if length is even, go to **Step** 5.
Else go to **Step** 6.
Step 5: Calculate key (K) for substitution by looking for Letter at Position n/2, Calculate a ASCII value By considering NUL(null)=0, SOH(start of header)=2…..DEL(delete)=127 and go to **Step** 7.
Step 6: calculate Key (K) for substitution by looking for Letter at Position (n+1)/2, Calculate a ASCII value by considering NUL(null)=0, SOH(start of header)=2…..DEL(delete)=127.
Step 7: Apply Substitution using formula C=(p+k) mod 254; where c is cipher text , p is plain text and key K.
Step 8: Apply transposition, i.e. Rail Fence Technique on result of **Step** 7 and go to **Step** 15.
Step 9: Apply rail-fence transposition to the Plain Text.
Step 10: Count the length of string.
Step 11: if length is even, go to **Step** 12.
Else go to **Step** 13.
Step 12: Calculate key (K) for substitution by looking for Letter at Position n/2, Calculate ASCII value by considering NUL(null)=0, SOH(start of header)=2 …..DEL(delete)=127 and go to **Step** 14.
Step 13: calculate Key (K) for substitution by looking for Letter at Position (n+1)/2, Calculate a ASCII value by considering NUL(null)=0, SOH(start of header)=2 …..DEL(delete)=127.
Step 14: Apply Substitution to the result of **Step** 9 using formula C=(p+k) mod 254; where c is cipher text , p is plain text and key K.
Step 15: On the transposed result, apply Logical gate NOT.
STOP
Final Cipher text will be the output of previous Step.

V. Encryption And Decryption Result

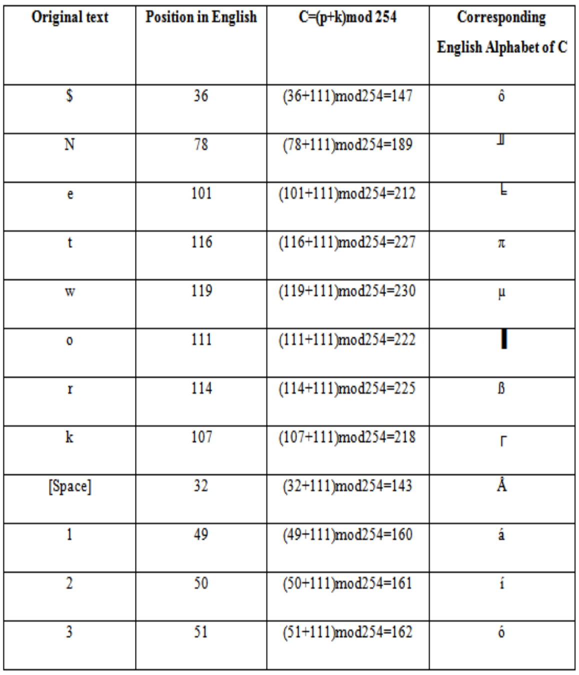
Example: Consider the plaintext message "$Network 123". The Encryption and Decryption results produced by the Algorithm are as follows

- A Random key is generated by random function.Let, key generated be 8. Since the key is even, substitution technique will be applied first or else transposition technique would have been applied first.
- Now the length of original string "$Network 123" is counted, which is 12.

Since 12 is an even number, the Key is generated using (n/2) i.e. 12/2=6 or else the key generated should be (n+1)/2.

Now the key i.e. letter at position 6 is 'o', and key chosen will be word's corresponding ASCII decimal value i.e. **k=111** (consider NUL (null) =0, SOH (start of header) =2 …..DEL (delete) =127).

Table I. Encryption



Original text	Position in English	C=(p+k)mod 254	Corresponding English Alphabet of C
$	36	(36+111)mod254=147	ô
N	78	(78+111)mod254=189	╜
e	101	(101+111)mod254=212	╘
t	116	(116+111)mod254=227	π
w	119	(119+111)mod254=230	µ
o	111	(111+111)mod254=222	▐
r	114	(114+111)mod254=225	ß
k	107	(107+111)mod254=218	┌
[Space]	32	(32+111)mod254=143	Å
1	49	(49+111)mod254=160	á
2	50	(50+111)mod254=161	í
3	51	(51+111)mod254=162	ó

After round 1 of encryption, we get -

ô╜ ╘πµ ▐ß ┌Åáíó.

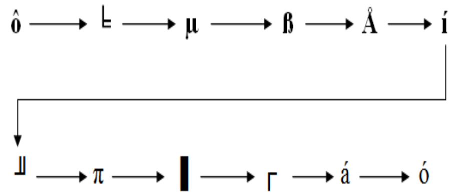


Fig.1 Apply transposition i.e. Rail Fence.

Cipher Text= **ô ╘µßÅí╜ π ▐ ┌áó**
In final Round of Encryption, apply Logical Gate 'NOT'.

Table II. Encryption

Original Text	ASCII Value	Binary Equivalent	NOT	Decimal Equivalent	ASCII Equivalent In Character
ô	147	10010011	01101100	108	l
╘	212	11010100	00101011	43	+
µ	230	11100110	00011001	25	↓
ß	225	11100001	00011110	30	▲
Å	143	10001111	01110000	112	p
í	161	10100001	01011110	94	^
╜	189	10111101	01000010	66	B
π	227	11100011	00011100	28	∟
▐	222	11011110	00100001	33	!
┌	218	11011010	00100101	37	%
á	160	10100000	01011111	95	_
ó	162	10100010	01011101	93	]

Final cipher text	l	+	↓	▲	p	^	B	∟	!	%	_	]

Table III. Decryption

	ASCII	ASCII's Binary	NOT	Decimal	ASCII
l	108	01101100	10010011	147	ô
+	43	00101011	11010100	212	╘
↓	25	00011001	11100110	230	µ
▲	30	00011110	11100001	225	ß
p	112	01110000	10001111	143	Å
^	94	01011110	10100001	161	í
B	66	01000010	10111101	189	╜
∟	28	00011100	11100011	227	π
!	33	00100001	11011110	222	▐
%	37	00100101	11011010	218	┌
_	95	01011111	10100000	160	á
]	93	01011101	10100010	162	ó

Text Obtained – **ô ╘µßÅí╜ π ▐ ┌áó**

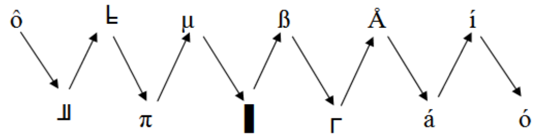


Fig.2 Apply reverse transposition as

Text Obtained: - **ô╜ ╘πµ ▐ß ┌Åáíó**
Key used in Encryption, k=111

Table IV:-Decryption

	Decimal Equivalent	P=|c-k| mod 254	Corresponding Alphabet in English
ô	147	|147-111|mod254=36	$
╜	189	|189-111|mod254=78	N
╘	212	|212-111|mod254=101	e
π	227	|227-111|mod254=116	t
µ	230	|230-111|mod254=119	w
▐	222	|222-111|mod254=111	o
ß	225	|225-111|mod254=114	r
┌	218	|218-111|mod254=107	k
Å	143	|143-111|mod254=32	[Space]
á	160	|160-111|mod254=49	1
í	161	|161-111|mod254=50	2
ó	162	|162-111|mod254=51	3

VI. Advantages Of Algorithm

- Less time complexity.
- Easy to understand and implement program.
- Uses basic and easy encryption schemes.
- Efficient key generation technique.
- ASCII value is used during encryption.
- High security because of the use of random number generator.

VII. Conclusion

The main aim of encryption is to convert the text into such a form that its crypt analysis becomes tedious and confusing. The algorithm provides good encryption and is automated. The keys used are very random and cannot be identified. And all this is achieved with simple and compact code which does not lead to large processing delay and time complexity. This leads to the high security in which data cannot be easily interpreted in the transfer of the message. In future the algorithm can also be applied to the Digital Image Processing and can be used to distort an image file and on the other hand the original picture can be retained.

REFERENCES

[1] Govind Prasad Arya, Aayushi Nautiyal, Ashish Pant, Shiv Singh & Tishi Handa, *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, Vol. 1, No. 1, March-April 2013.

[2] BRUE SCHNEIER, *Applied Cryptography Protocols, Algorithms and Source Coding*, John Wiley & Sons, Inc., Second Edition.

[3] IBM. (1994). *The Data Encryption Standard (DES) and its strength against attacks*. IBM Journal of research and Development, Vol. 38, PP. 243-250.

[4] Jonathan Katz, Yehuda Lindell Chapman & Hall, *Introduction to Modern Cryptography*.

[5] Rick Burgess (2011), URL: http://www.techspot.com / 52011-one-minute-on-the-internet-640tb-data-transferred-100k-tweets-204-million-e-mails-sent.html.

[6] R.Venkateswaram, Dr.V.Sundaram, (2010), *Information Security: Text Encryption and Decryption with Poly Substitution method and combining features of cryptography*.

[7] S. G. Srikantaswamy, H. D. Phaneendra (2011), *A Cipher Design using the Combined Effect of Arithmetic and Logic Operations with Substitutions and Transposition Techniques*, International Journal of Computer Applications (0975 – 8887) Volume 29– No.8.

[8] URL: www.asciitable.com.

[9] V. U. K. Sastry, D. S. R. Murthy, Dr. S. Durga Bhavani *A block cipher having a key on one side of plaintext Matrix and its Inverse on the other side*.

[10] William Stallings, *Cryptography and Network Security: Principles and Practice*, Pearson Education, Fifth Edition

[11] William Stallings, (2004). *Network Security Essentials (Applications and Standards)*, Pearson Education, pp. 2–80.