www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume - 3 Issue - 8 August, 2014 Page No. 7596-7576

Multi-Level Encryption using SDES Key Generation Technique with Genetic Algorithm

S.Devi[1], Dr.V.Palanisamy[2]

[1]Department of Computer Science, Alagappa University,
Karaikudi, Tamilnadu, India
Sakthies09@gmail.com

[2]Department of Computer Science, Alagappa University,
Karaikudi, Tamilnadu, India
vpazhanisamy@yahoo.co.in

Abstract: ***Data transmission in real time environment has been invoked by trustworthy persons as well as in internet media it resides on secure communication channel. The introduction of internet increases security issue twice for exchanging the information electronically. Cryptography is the process of scrambling the data into unknown format. This process is done with the help of encryption & decryption algorithm. The basic two ideas behind the cryptographic techniques are substitution & transposition. The Caesar cipher substitution method is used to alter the plaintext characters with the help of automatic key. This paper presents a multi stage encryption algorithm. At the end of each stage an intermediate cipher is produced. The transposition is employed by using crossover method of genetic algorithm. Final ciphertext is derived from the combined effect of basic arithmetic & logic operations.***

Keywords: SDES key generation, substitution, transposition, crossover, mutation, encryption and decryption.

1. Introduction

Cryptography as the art of writing or solving codes. First, it focuses solely on the problem of secret communication. Second, the definition refers to cryptography as an art form. Constructing good code or breaking existing ones, relied on creativity and personal skill.

Modern cryptography is the scientific study of techniques for security digital information and distributed computations. It concerned with the construction of ciphers (called encryption scheme) for providing secret communication between two parties sharing some information in advance

The setting in which the communication parties share some secret information in advance is known as the private-key (or the symmetric) setting. In this setting, the same key is used to convert the plaintext into a ciphertext and back. An asymmetric encryption setting involves the sender and receivers don't share any secrets & different keys are used [1].

1.1 Substitution and Transposition Ciphers

Substitution cipher is a method of encoding by which units of plaintext are replaced with ciphertext, according to a regular system; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing an inverse substitution. There are a number of different types of substitution cipher. If the cipher operates on single letters, it is termed a simple substitution cipher; a cipher that operates on larger groups of letters is termed polygraphic.

A transposition cipher is methods of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext [2].

1.2 Genetic Algorithm

The proposed algorithm uses the crossover and mutation concepts of genetic algorithm. Generally, crossover technique generates new individuals (offspring or children) from the two given parents (two individuals). Mutation operation randomly changes characters in the offspring produced from the crossover technique [3].

2. Related Work

2.1 Background Study

Several encryption algorithms have been made in the field of cryptography. To review this history, most of the encryption algorithm uses either the concept of an automatic key generation algorithm or combination of substitution and transposition methods.

S.G.Srikantasamy and H.D.Phaneedra in the year 2011 [4] presented an encryption algorithm based on the combination of arithmetic and logic operations. A private key is produced from the plaintext message itself. The Caesar or Shift cipher is the traditional or simple substitution cipher in the cryptography. To improve the Caesar cipher model, a modified approach to the Caesar cipher is utilized in the year 2012 by the same authors [5].

An automatic key generation concept is introduced in the year 2013 by B.Bazith Mohammed [6] to the Caesar cipher. It takes a single round for the encryption/decryption process.

Various ciphers are available to encrypt the plaintext characters. Apart from the Caesar cipher, one time pad cipher is used with the arithmetic and logic operations. S.G.Srikantasamy and H.D.Phaneedra proves the flexible key generation algorithm helps to improve the network security issues of plaintext messages [7].

Govind Prasad Arya and his team presents a new concept of using all techniques combined together into one encryption algorithm. But the algorithm has take only 26 values of alphabets for the plaintext messages [8]. The key value should be increased up to 254 ASCII values by Devendra Prasad & his team [9].

2.2 Existing SDES Key Generation Algorithm

First, permute the key in the following fashion. Let the 10-bit key be designated as (k1, k2, k3, k4, k5, k6, k7, k8, k9, k10). Then the permutation P10 is defined as:
P10 (k1, k2, k3, k4, k5, k6, k7, k8, k9, k10) = (k3, k5, k2, k7, k4, k10, k1, k9, k8, k6)
P10 can be concisely defined by:
P10 (k3, k5, k2, k7, k4, k10, k1, k9, k8, k6)
This P10 is read from left to right; each position in the P10 gives the identity of the input bit that produces the output bit in that position. So the first output bit is bit 3 of the input; the second output bit is bit 5 of the input, and so on.
For example, the key (1010000010) is permuted to (1000001100).
Next, perform a circular left shift (LS-1), or rotation, separately on the first five bits and the second five bits.
In our example, the result is (00001 11000).
Next we apply P8, which picks out and permutes 8 of the 10 bits according to the following rule:
P8 (6, 3, 7, 4, 8, 5, 10, 9)
The result is subkey1 (K1). In our example, this yield (10100100) we then go back to the pair of 5-bit strings produced by the two LS-1 function and perform a circular left shift of 2 bit positions on each string.
In our example, the value (00001 11000) becomes (00100 00011).
Finally, P8 is applied again to produce K2.
In our example, the result is (01000011).
K1=10100100; K2=01000011[10].
K1=164, K2=67

3. Encryption and Decryption Algorithm

3.1 Encryption Algorithm

The key deduced from SDES key generation algorithm have a couple of key values i.e. k1=164 & k2=67. To find the best one, chosen is made by pick out the smallest key value among them. Table 1[5] lists the frequently used characters & its values for message and key values. The reason for taking the smallest key value is determined from the Caesar cipher method. The fact that has been defined in Caesar cipher is the key value must be small when compared to the message length.

- An automatic key is devised through the existing SDES key generation algorithm. In the first stage, Caesar cipher substitution is performed with the help of private key produces an intermediate cipher1.
- Takes the crossover of characters on the resultant intermediate cipher1. To transpose the character's position, mutation process is involved. An intermediate cipher2 is produced at the end of second stage.
- The final stage encompasses all basic arithmetic and logic operations yields final ciphertext. The ciphertext generated from the encryption algorithm is in disguised format because of employing the multi stage processes.

Table 1: Frequently used Characters and its Values

Character	*Value*	*Character*	*Value*
!	0	V	53
“	1	W	54
#	2	X	55
$	3	Y	56
%	4	Z	57
&	5	[	58
‘	6	\	59
(	7	]	60
)	8	^	61
*	9	_	62
+	10	`	63
‘	11	a	64
-	12	b	65
.	13	c	66
/	14	d	67
0	15	e	68
1	16	f	69
2	17	g	70
3	18	h	71
4	19	i	72
5	20	j	73
6	21	k	74
7	22	l	75
8	23	m	76
9	24	n	77
:	25	o	78
;	26	p	79
<	27	q	80
=	28	r	81
>	29	s	82
?	30	t	83
@	31	v	85
A	32	w	86
B	33	x	87
C	34	y	88
D	35	z	89
E	36	{	90
F	37	|	91
G	38	}	92
H	39	~	93
I	40		
J	41		
K	42		
L	43		
M	44		
N	45		
O	46		
P	47		
Q	48		
R	49		
S	50		
T	51		
U	52		

3.2 Example for Encryption Algorithm

3.2.1 Round 1:

Plaintext = Encipher, Key=67 (Private key)

Table 2: Substitution Method

Original Text	*Numeric Values of English Alphabet*	*C=(P+k) mod 94*	*Corresponding English Alphabet of C*
E	36	9	*
n	77	50	S
c	66	39	H
i	72	45	N
p	79	52	U
h	71	44	M
e	68	51	J
r	81	54	W

At the end of round 1, Intermediate cipher 1 is *SHNUMJW

3.2.2 Round 2:

*	S	H	N
U	M	J	W

Crossover
St1 = *MHW
St2 = USJN
St=st1+st2
St=*MHWUSJN
Mutation
M=inv (st)
M=NJSUWHM*
At the end of round 2, Intermediate cipher 2 is NJSUWHM*

3.2.3 Round 3:

Table 3: Combination of Arithmetic and Logic Operations

Intermediate cipher 2	*Numerical Value*	*Binary Equivalent*	*Left Shift by 2*
N	45	00101101	10110100
J	41	00101001	10100100
S	50	00110010	11001000
U	52	00110100	11010000
W	54	00110110	11011000
H	39	00100111	10011100
M	44	00101100	10110000
*	9	00001001	00100100

Table 4: Process 2

Complement	*Decimal Equivalent*	*ASCII Value*
01001011	75	K
01011011	91	[
00110111	55	7
00101111	47	/
00100111	39	|
01100011	99	C
01001111	79	O
11011011	219	█

At the end of round 3, Final ciphertext is K [7/|cO█

3.3 Decryption Algorithm

Decryption algorithm is just the reverse process of an encryption algorithm.

- Take the ciphertext as the input and perform all basic arithmetic and logic operations to it. It produces the intermediate plaintext 1.
- Apply the crossover process to the intermediate plaintext 1 followed by the mutation technique which gives the intermediate plaintext2.
- Finally, reverse Caesar cipher substitution is involved using private key as declared in the encryption algorithm.

3.4 Example for Decryption Algorithm

3.4.1 Round 1:

Ciphertext= K[7/|cO █,Key=67 (Private key)

Table 5: Combination of Arithmetic and Logic Operations

Cipher Text	*ASCII Value in Decimal*	*Binary Equivalent*	*Right shift by 2*
K	75	01001011	11010010
[	91	01011011	11010110
7	55	00110111	11001101
/	47	00101111	11001011
|	39	00100111	11001001
C	99	01100011	11011000
O	79	01001111	11010011
█	219	11011011	11110110

Table 6: Process 2

Complement	*Decimal Equivalent*	*Intermediate plaintext 1*
00101101	45	N
00101001	41	J
00110010	50	S
00110100	52	U
00110110	54	W
00100111	39	H
00101100	44	M
00001001	9	*

At the end of round 1, Intermediate plaintext1 is NJSUWHM*

3.4.2 Round 2:

Mutation
M=inv (intermediate plaintext 1)
M=*MHWUSJN
Crossover

*	M	H	W
U	S	J	N

St1 =*SHN
St2 = UMJW
St = St1 + St2
St= *SHNUMJW
At the end of round 2, Intermediate plaintext 2 is *SHNUMJW

3.4.3 Round 3:

Table 7: Reverse Substitution Method

Intermediate plaintext 2	Numerical Value	C=(P-k) mod 94	Original plaintext
*	36	9	E
S	77	50	n
H	66	39	c
N	72	45	i
U	79	52	p
M	71	44	h
J	68	51	e
W	81	54	r

At the end of round 3, Original plaintext is retrieved
Original plaintext = Encipher.

3.5 Merits of the Proposed Algorithm

- Suitable for maximum frequently used characters and numerical values.
- There are 94! attempts are possible for the cryptanalytic attack.
- High processing speed and low process delay.
- Combination of techniques and genetic programming concepts makes the cryptanalysis difficult.
- Brute force attack is certainly feasible for the proposed automatic key.

4. Experimental Result

4.1 Comparisons with Existing Algorithms

The following table shows relationship between existing symmetric cryptographic algorithms and proposed encryption algorithm. The proposed algorithm consists of variable input size and key sizes as well as 2^{256} alternate keys are possible for the different input size. The encryption/decryption process defined in this paper is more secure against threats and hackers.

Table 8: Comparison of Various Symmetric Cryptographic Algorithms

Algorithm	Key Size(bits)	Input Size(bits)	Number of Alternate Keys
DES	56	64	256
AES	128	128	2^{128}
Triple DES	168	128	2^{168}
Blowfish	Variable	Variable	2^{256}
RC5	Variable	Variable	2^{256}
RC4	Variable	Variable	2^{256}
Proposed Algorithm	Variable	Variable	2^{256}

4.2 Simulation Analysis

For the propose work, Intel Core i3-3120M of CPU speed 2GB RAM is used. In this experiment the input sizes range from 110 bits to 175 bits against time in minutes in first chart. The performance metrics are analyzed by the Encryption and decryption time.

The calculation and analysis purpose for the proposed algorithm, customized computer application program is developed in C#.NET platform and after execution for analysis purpose the data is shown in MS Excel from there we can direct create graphs for visual analysis.

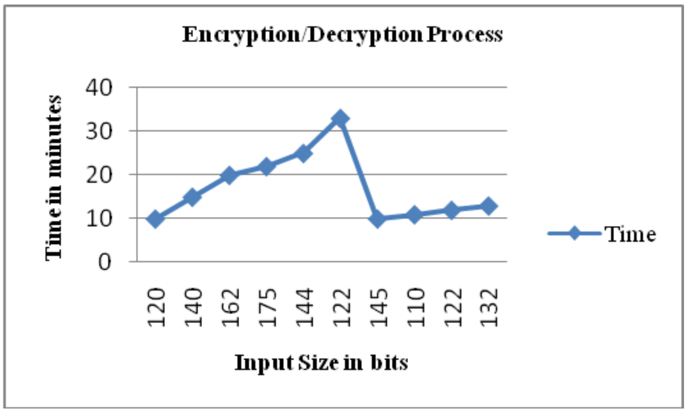

Figure 1: Execution Time for Encryption/Decryption Process

5. Conclusion

The effort of the proposed algorithm is to make the cryptanalysis difficult and algorithm stronger. In the previous encryption algorithm, key value is generated from the message itself. But this paper presents a multi stage encryption algorithm which is efficient for 94 character values. Using an automatic key value for encrypting characters of plaintext hides the relationship between the ciphertext & plaintext. An intermediate cipher resulted from each stage is considered to be an input of the next stage. So the final ciphertext is in tedious format to analyze. This algorithm is applied to maximum of length for the message and key values. Whenever the key is long, the encipherment is must strong against the intruders. In the future work, this algorithm will be extended to asymmetric key cryptographic system with random key generation technique.

References

[1] Jonathan Katz and Yehuda Lindell, Introduction to modern cryptography, Chapman & Hall/CRC, Taylor & Francis Group, 2008.
[2] URL: http://en.wikipedia.org/wiki/Transposition_cipher
[3] URL: http://www.myreaders.info/09_Genetic_Algorithms.pdf
[4] S.G.Srikantaswamy and Dr.H.D.Phaneendra," A cipher design using the combined effect of arithmetic and logic operations with substitutions and transposition techniques," International Journal of Computer Applications (0975-8887), vol.29, no.8, pp.34-36, Sep.2011.
[5] S.G.Srikantaswamy and Dr.H.D.Phaneendra," Improved Caesar cipher with random number generation technique and multistage encryption," International Journal on Cryptography and Information Security (IJCIS), vol.2, no.4, pp.39-49, Dec. 2012.
[6] B. Bazith Mohammed," Automatic Key Generation of Caesar Cipher," International Journal of Engineering Trends and Technology (IJETT), vol. 6, no. 6, pp.337-339, Dec. 2013
[7] S.G.Srikantaswamy and Dr.H.D.Phaneendra,"Enhanced onetime pad cipher with more arithmetic and logical operations with flexible key generation algorithm," International Journal of Network Security & Its Applications (IJNSA), vol.3, no.6, pp. 243-248, Nov.2011.
[8] Govind Prasad Arya, Aayushi Nautiyal, Ashish Pant, Shiv Singh & Tishi Handa, " A cipher design with automatic key generation using the combination of

substitution and transposition techniques and basic arithmetic and logic operations,"The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), vol. 1, no. 1, pp. 21-24, Mar. 2013.

[9] Devendra Prasad,Govind Prasad Arya,Chirag Chaudhary,Vipin Kumar,"Encipher A text encryption International Journal of Computer Science and Information Technologies (IJCSIT), vol. 5 (2) , 2014.

[10] William Stallings, Cryptography and network security, 3rd ed., Pearson Education, Prentice Hall, 2007.