Design and Implementation of Cipher Algorithm using Randomized Alphanumeric Characters

Sudhakar Kumar Singh

School of Computing Science and Engineering, VIT University, Vellore - 632014,
TamilNadu India

Abstract- Cryptography is the technique that is used to ensure the safety of communication over the network in most of the computer communication systems. The strength of a cryptographic algorithm is based on the difficulty of cryptanalysis imposed over system. Our proposed Algorithm provides the best security analysis, in terms of computational time and effective environment over the network. As we are aware of the security of communication has attracted the attention towards the design of new Algorithm. It is a combination of some of the best known encryption algorithms in existence. Here, we apply the concept of RAC (Randomized Alphanumeric Characters) proceeding with some computation to reach the goal of cipher text. The algorithm that we are proposing in our work is the resultant of some existing algorithms that uses the strengths of one algorithm to compensate the weakness of other by applying RAC as well as applying a technique which is similar to Bit-Stuffing**.**

Keyword- Cryptography, Security, RAC, cipher text.

I. Introduction

The concept of cryptography begins thousands of years ago; it takes number of versions based on the standards and the analysis which ensures the standards of cryptography. Encryption is the technique of changing the format of plaintext to cipher text which is unrecognizable and useless to unauthorized party. Decryption is the just reverse of encryption. It can be done in three ways (I) Secret key (symmetric): uses a single key for both encryption and decryption (II) Public key (asymmetric): uses two keys one for encryption and other for decryption. (III) Hash function (one way cryptography). In our proposed algorithm we are using concept of XOR operation and bit stuffing in a specific way.

II. Existing Algorithms:

Several algorithms are available for the cryptography like RSA, IDEA etc all these are used for the purpose of data security. Each and every algorithm has its own public keys and private keys with some special features and may use different mathematical concepts to maintain the standards of cryptography.

RSA: RSA is based on Public Key and Private Key, where Public Key is made known to others while private key must be kept in secret.

The step below shows how RSA keys are generated.

1. Two large prime numbers, p and q are chosen.

2. Compute the product of these two N=p.q.

3. Compute $\varphi (N) = (p - 1) (q - 1)$

4. e is the public key, chosen randomly

5. Find private key, d. d must be satisfy

$e.d = 1 \bmod \varphi (N)$

For encryption: - Cipher, $c = me \bmod \varphi (N)$

For decryption: - Message, $m = ce \bmod \varphi (N)$

IDEA: IDEA is based on public key.

The steps below show how IDEA key are generated.
1. Original text is divided into 64-bit blocks.
2. Each 64-bit block is further divided into four 16-bit sub-blocks: X1, X2, X3, X4.
3. The 128-bit IDEA session key is divided into eight 16-bit key-blocks: Ki, 1, Ki, 2, Ki, 3, Ki, 4, Ki, 5, Ki, 6, Ki, 7, Ki, 8.
4. Addition and Multiplication are perform on each block of Xn and Ki, j.
5. The combination of operations is performed eight times to get the final encryption.

III. Proposed Algorithm

Our proposed algorithm uses Randomized Alphanumeric Character and applying XOR operation with message and the technique of Bit-stuffing in specific way. We used asymmetric private key and the public key which increases the efficiency level of encryption algorithm .Here one 8-bit public key is randomly selected among several generated Alphanumeric Characters and one 5-bit private key. These may vary in real life environment.

Encryption Flowchart of our proposed algorithm is given below.

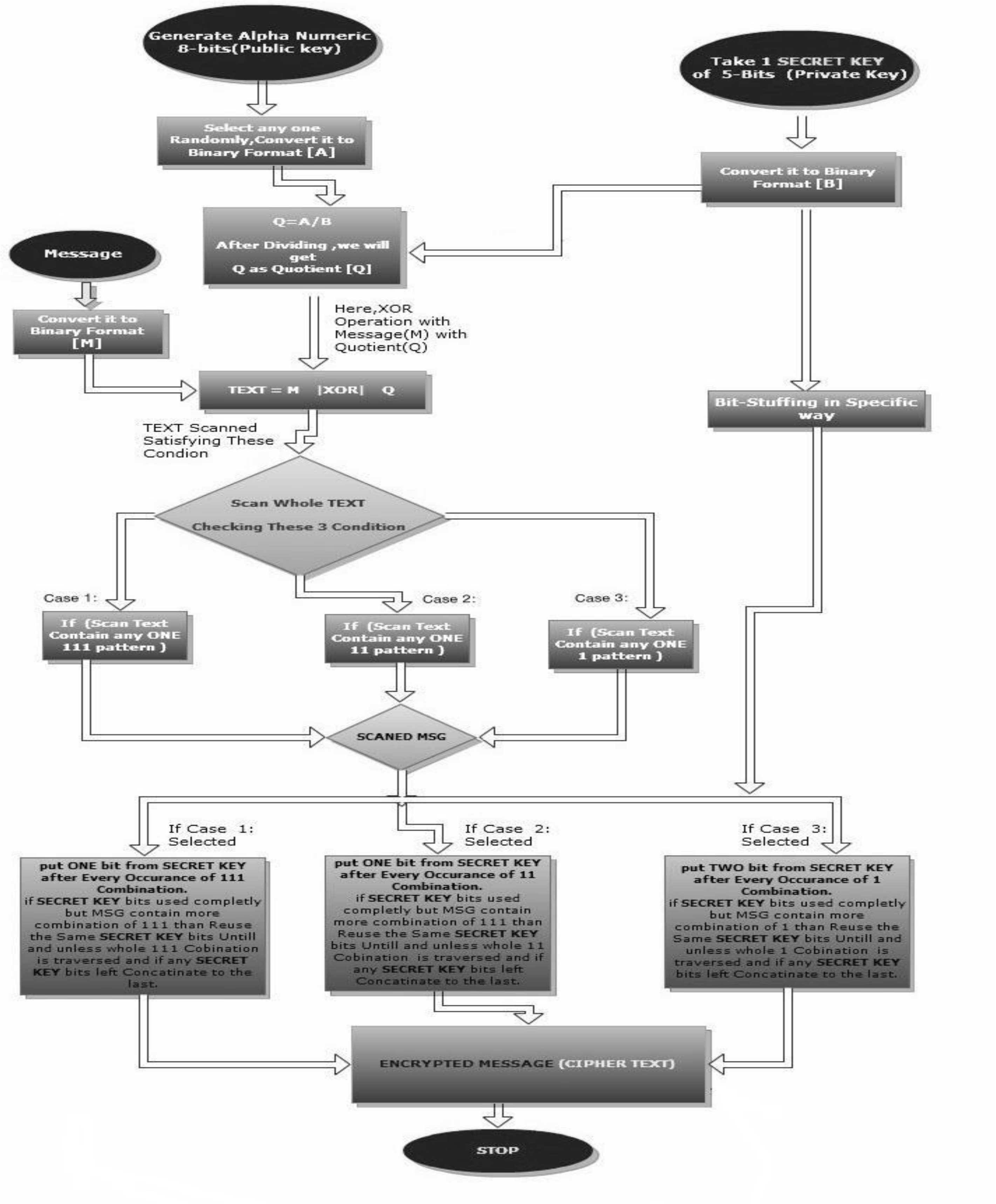
Generate Alpha Numeric 8-bits(Public key)
Take 1 SECRET KEY of 5-Bits (Private Key)
Select any one Randomly,Convert it to Binary Format [A]
Convert it to Binary Format [B]
Q=A/B
After Dividing ,we will get Q as Quotient [Q]
Message
Convert it to Binary Format [M]
Here,XOR Operation with Message(M) with Quotient(Q)
TEXT = M |XOR| Q
Bit-Stuffing in Specific way
TEXT Scanned Satisfying These Condion
Scan Whole TEXT
Checking These 3 Condition
Case 1:
Case 2:
Case 3:
If (Scan Text Contain any ONE 111 pattern)
If (Scan Text Contain any ONE 11 pattern)
If (Scan Text Contain any ONE 1 pattern)
SCANED MSG
If Case 1: Selected
If Case 2: Selected
If Case 3: Selected
put ONE bit from SECRET KEY after Every Occurance of 111 Combination. if SECRET KEY bits used completly but MSG contain more combination of 111 than Reuse the Same SECRET KEY bits Untill and unless whole 111 Cobination is traversed and if any SECRET KEY bits left Concatinate to the last.
put ONE bit from SECRET KEY after Every Occurance of 11 Combination. if SECRET KEY bits used completly but MSG contain more combination of 111 than Reuse the Same SECRET KEY bits Untill and unless whole 11 Cobination is traversed and if any SECRET KEY bits left Concatinate to the last.
put TWO bit from SECRET KEY after Every Occurance of 1 Combination. if SECRET KEY bits used completly but MSG contain more combination of 1 than Reuse the Same SECRET KEY bits Untill and unless whole 1 Cobination is traversed and if any SECRET KEY bits left Concatinate to the last.
ENCRYPTED MESSAGE (CIPHER TEXT)
STOP

Fig 1. Encryption flowchart

IV. Proposed Algorithm Process

Encryption Algorithm

1. Generate N number of 8-bit Alphanumeric Characters randomly.

2. Randomly select any one among them, denoted as ‘A’, and 5-bit SECRET KEY assigned as ‘B’.

3. Convert ‘A’ and ‘B’ in binary format.

4. Compute the quotient of these two Q=A/B.

5. Perform XOR operation with Message ‘M’ and quotient ‘Q’, TEXT = M XOR Q.

6. Scanned TEXT and perform the Bit-Stuffing as per flowchart.

7. In this way we get the encrypted message as CIPHERTEXT.

Decryption Algorithm

1. Remove the extra bit from CIPHERTEXT.
2. Perform the XOR operation with ‘TEXT’ and quotient ‘Q’.
3. In this way we get the original message ‘M’.

V. Working Example

Encryption:

1. Input: Public Key as ‘A’=SCSE04, Secret Key as ‘B’=VIT and Message as ‘M’=11MSC

2. Conversion to binary format:

Message as M=11MSC
[1=0000001, 1=0000001, M=1001101, S=1010011, C=1000011]
M=00000010000001100110110100111000011 (11MSC)

Public key A= SCSE04
[S=1010011, C = 1000011, S= 1010011, E=1000101, 0= 0000000, 4= 0000100]
A= 101001110000111010011100010100000000000100 (SCSE04)

Secret key B=VIT
[V=1010110, I=1001001, T=1010100]
B=101011010010011010100 (VIT)

3. Quotient ‘Q’; Q=A/B=1111011011111110110001

4. TEXT = M XOR Q.
TEXT=10000001011011011011110011.

5. CIPHERTEXT = TEXT with Bit-stuffing using SECRET KEY, (Case: 2)
CIPHERTEXT = 100000010111011001110110001111010010011010100
[Encrypted Message (CIPHERTEXT) = 100000010111011001110110001111010010011010100]

Decryption:

It is just reverse of encryption.

1. Input as CIPHERTEXT = 100000010111011001110110001111010010011010100

Case 1: Remove the bit after 111 patterns and concatenate.
Removed bits =001010010011010100
Comparison of removed bits with SECRET KEY bits, here mismatched, so break and go to next case.

Case 2: Remove the bit after 11 patterns and concatenate.
Removed bits=101011010010011010100
Comparison of removed bit with SECRET KEY bits, here it matched, so go to next phase.

2. Performing XOR operation with TEXT and Quotient. In this manner we get the original message.
Message M = 00000010000001100110110100111000011 (11MSC)

VI. Conclusion

Overall RSA, DES and IDEA are very strong encrypting algorithms in existence, they do have their weaknesses: RSA contains lengthy and complex computations, while the purpose of initial and final permutation is not clear in DES algorithm, whereas IDEA uses a single and lengthy key. By combining the concept of above mentioned algorithms, our proposed algorithm uses the strength of these two algorithms to in order to reduce the weaknesses. As the result, our proposed cipher algorithm is one of the strongest, simplest and fastest encryption algorithms.

References

[1] S.C. Coutinho,University Press (India) Private Limited (2003).The Mathematics of Ciphers ,Number Theory and RSA Cryptography. Department of Computer science Federal University of Rio de Janeiro Rio de Janerio, Brazil

[2] Evolution of Cryptography Mohd Zaid Waqiyuddin Mohd Zulkifli, Evolution of Cryptography, 17th January 2007.

[3] Data Encryption and Decryption process Using Bit Shifting and Stuffing (BSS) Methodology, B.Ravi Kumar et al. / International Journal on Computer Science and Engineering (IJCSE).

[4] Wikipedia, The Free Encyclopedia, from http://en.wikipedia.org/wiki/RSA

[5] Wikipedia, the Free Encyclopedia, from

[6] http://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm

[7] Wikipedia, The Free Encyclopedia, from http://en.wikipedia.org/wiki/Data_Encryption_Standard

[8] CareerRide.com, from http://www.careerride.com/Networking-DES-weakness-and-strength.aspx

Authors

First Author – Sudhakar Kumar Singh, M.SC (Computer Science), VIT University Vellore, Sksingh2012@yahoo.com.

Second Author – Hariom Kumar, M.SC (Computer Science)

Third Author – K Praveen Kumar, M.SC (Computer Science)

Fourth Author – K Ramesh Babu, Faculty of VIT University. Vellore.