A New Method For Generation of Variable Session Keys

Authors

Akhand pratap singh¹, Suneel Kumar²

(Asst. Prof. UCER Allahabad)

Email – rakhand313@gmail.com

ABSTRACT

In all cryptosystem the key challenge is to make key unbreakable. Shannon proposed that key would be impossible to break if the key is made time variant. The time variant key can be implemented by changing key from session to session. Also, it is essential to have key which is variable from session to session for achieving perfect security. To achieve this, key should be made variable by an agreement that creates new key for each session at both sender and receiver sides. So, we propose a new technique for generating time variant key. The purpose of this algorithm is to produce sequence of numbers whose behavior is very hard to distinguish from that of their “truly random” counterparts. Key requirements may differ on the context. It ensures better randomness among the successive two session keys. The new technique can be applied in encryption/decryption techniques like RSA, DES, and AES.

Keywords – Perfect Security, Truly Random, Randomness, RSA, DES, AES

INTRODUCTION

Information security has a very important role nowadays. The requirement of information security is increasing because of widespread use of distributed systems, network and communication facilities for carrying information between terminal user and computer and between computer and computer. Hence to provide confidentiality authentication, integrity and non-repudiation, information security has evolved.

Large number of algorithms and techniques are designed for secure transmission of data. Cryptographic algorithms play a central role in information security systems. There are two general types of key-based algorithms: Symmetric and Asymmetric algorithms. Symmetric algorithms (also called secret-key algorithms) are algorithms where the encryption key can be calculated from the decryption key and vice versa. In symmetric algorithms, the encryption and decryption key are the same. Both the sender and receiver agree on a key before they can communicate securely. On the other hand, in Asymmetric algorithms (also called public key algorithms) the decryption key cannot be calculated from the encryption key. So,
keys play an important role in the security of any cryptographic algorithm. If weak key is used in algorithm, then any intruder may decrypt the data. One of the central factors contributing to the strength of symmetric key algorithms is the size of key used. In practice, most good cryptographic algorithms rely on increasing the key size to strengthen the security of algorithm.

But a better way to increase level of security is to make keys time variant. Time variant key can be implemented by changing key from session to session. So we propose a new algorithm based on varying the key to increase the security of cryptosystem. The purpose of this algorithm is to produce a sequence of keys whose behavior is very hard to predict.

ALGORITHM FOR KEY GENERATION

The idea of time variant key, namely Automatic Variable Key (AVK) has been introduced recently. In AVK new keys are generated for every session by an agreement between sender and receiver. The process of new key generation is as:

Let, \( K_0 \) is initial key which is exchanged between sender and receiver using any secret method. Then subsequent keys for different data to be exchanged are

\[
K_i = K_{i-1} \oplus D_{i-1} \quad \text{for } i > 0 \quad \text{(1)}
\]

A new key is generated for every new data value to be exchanged and used for encryption and decryption of that data value at both sender and receiver sides. Research on this AVK method has already been done in sufficient measure.

PROPOSED NEW METHOD OF AVK

The new method for AVK is as below:

Step 1. Initial key (K) and one random number (N) is exchanged between the sender and receiver by RSA.

Step 2. Now generate a new key as:

\[
K_i = K_{i-1} \oplus N \quad \text{(1)}
\]

Step 3. If \( N \) is ODD generate subsequent keys as:

\[
K_i = K_{i-1} \ll N \oplus D_{i-1} \quad \text{(2)}
\]

Where \( K_{i-1} \ll N = K_{i-1} \ll N \) (Circular left shift by \( N \) bits)

Else if \( N \) is EVEN generate subsequent keys as:

\[
K_i = K_{i-1} \oplus D_{i-1} \quad \text{(3)}
\]

Generate keys using eqn. (2) or (3) \( N \) times.

Step 4. Now exchange another random number (N) between sender and receiver by RSA.

Step 5. Subsequent key will be generated by same process as above starting from Step 2 and the process will repeat.

Example: If initial key exchanged = 1001 and \( N = 2 \) and \( D_0 = 1000 \)
Then First key = 1001 XOR 0010 = 1011
Now Second key = 1011 XOR 1000 = 0011
Third key = 0011 XOR 1001(D1) = 1010
A new random number N is exchanged between sender and receiver.
Let now N = 1
Then Fourth key = 1010 XOR 0001 = 1011
Now Fifth key = 0111 XOR 1010 (D2) = 1101
A new random number N is exchanged between sender and receiver.

ANALYSIS AND RESULT

For analysis we assume a parameter of randomness as a measure of amount of variation between the successive keys. So the randomness is defined as the number of bit locations in which any two successive keys vary. For example:

K_{i-1} = 10010111 and K_i = 11100001 then the randomness between these two successive keys is 5. So K_i is random to K_{i-1} by 5.

For the new method the initial key is \{11001010\}, initial random number is \{4\} and set of initial data is \{00000000,00000001,00000010,00000011,00000100,00000101,00000110,00000111,00001000,00001001,00001010,00001011,00001100,00001101,00001110,00001111\}.

The randomness as defined above is calculated by running a program and the result obtained is portrayed in figure.

![Randomness Graph](image)

Figure: Randomness of keys generated by new method

Figure 1: Randomness of keys generated by new method
CONCLUSIONS
After studying the AVK method of key generation, a new method has been proposed in this paper. This method is better than the AVK in case of randomness. The new method of key generation needs to be applied in encryption / decryption techniques.

The new method can be applied in AES. The effect of Brute force attack and Differential frequency attacks on the new technique needs to be examined. I propose to study these things in future work.

REFERENCES