Various New Methods of Implementing AVK

Rajat Subhra Goswami  
Department of Computer Science and Engineering,  
National Institute of Technology, Arunachal Pradesh-791112, India  
(Email: rajat.nitap@gmail.com)

Abhinandan Bhinia  
Microsoft Corporation, USA  
(Email: abhi1.bhunia@gmail.com)

Abstract—Researchers have proposed many data encryption techniques / standards to protect data from various types of attacks like brute force attack, frequency attack and differential frequency attack. Shannon [1-2] documented the theory of perfect secrecy with time variant key. An idea of time variant key namely Automatic Variable Key (AVK) was introduced by Bhinia [4-6]. The superiority of AVK was widely experimented by many researchers over a fixed or single key [7-14]. This paper proposed new protocols of Automatic variable Key (AVK) in cryptography.

Keywords-Perfect Security, RSA, DES, AVK, Computing & Shifting AVK (CSAVK), Decimal Shifting AVK (DSAVK), Randomness.

I. INTRODUCTION
To establish the superiority of time variant key in achieving perfect security is studied in [7-14]. The main challenge for the researchers / designers is to generate a key for producing the cipher document. The famous Vernum code was the first attempt to achieve perfect security but no effective variable key was applied or no concrete theory was established. The fundamental research of Shannon [1-2] in light of perfect security is that, the secret key will vary from session to session. An idea of time variant key, namely Automatic Variable Key (AVK) has been introduced by Bhinia [4-6]. Reasonable amount of research on AVK has been done elsewhere [7-14].

In AVK technique which illustrated in Table-1 the key is made variable by an agreement that creates new key for each data. This is reviewed as below:

Say, \( K_0 \) = initial key that may be exchanged by any conventional secret mode between a sender and a receiver. Subsequent keys for different data \( (D_{i-1}) \) to be exchanged are generated are:

\[
K_i = K_{i-1} \text{ XOR } D_{i-1} \quad \text{for } i \geq 0
\]

The key is made variable with exchanged data between a sender and a receiver. A new key is generated every time a data is exchanged. The new key so generated is used subsequently for further exchange of data. The illustrated technique of AVK has been extensively applied in both private and public key cryptography. The application is found to reduce brute force attack, frequency attack and differential frequency attack [4-10].

II. VARIOUS NEW IDEAS OF AVK TECHNIQUE
In CSAVK [11,13] technique illustrated in below the key is made variable by one agreement that also creates new key for each data.

Say, \( K_0 \) = initial key that may be exchanged by any conventional secret mode between a sender and a receiver. Subsequent keys for different data \( (D_{i-1}) \) to be exchanged are generated are:

\[
K_i = K_{i-1} \text{ XOR } D_{i-1} \quad \text{for } i \geq 0
\]

Where \( K_{i-1} = \text{ Bit wise right shifted (circular) } K_i \) / the number of shift will be the number of 1’s present in \( K_{i-1} \).

\( D_{i-1} = \text{ Bit wise left Shifted (Circular) } D_{i-1} / \) the number of shift will be number of 1’s present in \( D_{i-1} \).

In DSAVK [12] technique illustrated in below the key is made variable by another agreement that also creates new key for each data.

Initial key \( (K_0) \) is exchanged between the sender and the receiver.

Subsequent key, \( K_i \) (at \( i^{th} \) stage) is generated by both sender & receiver as:

\[
K_i = K_{i-1} \text{ XOR } D_{i-1} \quad \text{for } i \geq 0
\]

Where \( K_{i-1} = \text{ Bit wise right shift (Circular) } K_i / \) the number of shift will be the corresponding decimal value of \( K_{i-1} \).
The key is made variable with exchanged data between a sender and a receiver every time a data is exchanged. The new key so generated is used subsequently for further exchange of data.

Table 1: Elucidation of application of simple AVK in cryptology

<table>
<thead>
<tr>
<th>Session slots</th>
<th>Sender sends his /her private key to receiver</th>
<th>Receiver recovers his /her private key from sender</th>
<th>Receiver sends his /her private key to sender</th>
<th>Sender receives private key from receiver</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Secret key Say 5(101)</td>
<td>101</td>
<td>A secret key Say 7(111)</td>
<td>111</td>
<td>For next slot sender will use 111 as key and receiver 101 as key for transmitting data</td>
</tr>
<tr>
<td>2</td>
<td>Sender sends first (random 3) data 011 @ 111 =100</td>
<td>Receiver gets original data 011 @ 111 =011</td>
<td>Receiver sends first (random data 9) as 1001 @ 0101=1100</td>
<td>Sender sends back original data as 1001 @ 0101 =0101 =1001</td>
<td>Sender will create new key 0111 @ 1001 for next slot receiver will create new key 101 @ 011</td>
</tr>
<tr>
<td>3</td>
<td>Sender sends new data 4(100) as 0100 @ 0111 @ 1001</td>
<td>Receiver gets original data 0100 @ 0111 @ 1001 @ 0111 =0111</td>
<td>Receiver sends next data 8 (10000 @ 0101 @ 0011)</td>
<td>Receiver receives original data 1000 @ 0101 @ 0011 @ 0101 =0111 =1000</td>
<td>Sender computes new key 0111 @ 1001 receiver computes key 1001 @ 1000 for transmitting next data</td>
</tr>
</tbody>
</table>

III. NEW IDEAS

We propose various new ways for generation of key in AVK. The objective is to enhance the level of security by making more randomness between successive AVKs.

The new ideas for generation of key are as below:

A. PROTOCOL-I:

i) Initial key (K0) and one noise burst (m) is exchanged between the sender and the receiver by RSA.

ii) Subsequent key, Kj (at jth stage) is generated by both sender & receiver as:

\[ K_j = K_{j-1} \oplus D_{j-1} \] (AVK technique) for \( j \geq 0 \) .... (4)

iii) When X=m, another key (K_m) and another noise burst (n) is exchanged between the sender and the receiver.

iv) Subsequent key will be generated same way as eqn. 4 & process will repeat.

Example: In first case if \( k_{i-1} = 1001 \), \( D_{i-1} = 1000 \) and \( m=2 \), the subsequent keys will be:

Step 1: First key = 1001 \( \oplus 1000 = 0001 \)

Step 2: Next key = 0001 \( \oplus 1001(D_{j}) = 1000 \)

Step 3: New key and another noise burst will be exchanged between sender and receiver.

B. PROTOCOL-II:

i) Initial key (K0) and one noise burst (m) is exchanged between the sender and the receiver by RSA.

ii) Subsequent key, K_i (at ith stage) is generated by both sender & receiver as:

\[ K_{i+1} = K_i \oplus X \oplus D_i \] (CSAVK technique) for \( i \geq 0 \) .......... ...... (5)

iii) When X=m, another key (K_m) and another noise burst (n) is exchanged between the sender and the receiver.

iv) Subsequent key will be generated same way eqn. 5 & process will repeat.

Example: In first case if \( k_{i-1} = 1001 \), \( D_{i-1} = 1000 \) and \( m=2 \), the subsequent keys will be:

Step 1: First key = 0110 \( \oplus 0001 = 0111 \)

Step 2: Next key = 0110 \( \oplus 0110 (Assuming previous data = 1001) = 1000 \)

Step 3: New key and another noise burst will be exchanged between sender and receiver.

C. PROTOCOL-III:

i) Initial key (K0) and one noise burst (m) is exchanged between the sender and the receiver by RSA.

ii) Subsequent key, K_i (at ith stage) is generated by both sender & receiver as:

\[ K_i = K'_i \oplus X \oplus D'_i \] (DSAVK technique) for \( i \geq 0 \) .......... ...... (6)

iii) When X=m, another key (K_m) and another noise burst (n) is exchanged between the sender and the receiver.

iv) Subsequent key will be generated same way as eqn. 6 & process will repeat.

Example: In first case if \( k_{i-1} = 1001 \), \( D_{i-1} = 1000 \) and \( m=2 \), the subsequent keys will be:

Step 1: First key = 1111 \( \oplus 0100 = 1011 \)

Step 2: Next key = 1101 \( \oplus 0110 (Assuming previous data = 1001) = 1000 \)

Step 3: New key and another noise burst will be exchanged between sender and receiver.

IV. ILLUSTRATION OF AVK, CSAVK & DSAVK

A. ILLUSTRATION OF AVK
Let us assume that sender sends original data (D₀)00000100 in encrypted form using an initial key (K) = 10101010. Then in order to maintain the linearity, the encrypted form is 00000100 XOR 10101010 = 10101110.

At receiver end receiver will perform 10101110 XOR 10101010 and gets 00000100.

**B. ILLUSTRATION OF CSAVK**

Let sender sends initial data D₀(01011001) in encrypted form using key K(10101010). As per technique of CSAVK of eqn.(2) next key will be generated as K₀=00011000. The process will then be continued.

But in the next data transmission key will be changed by left shifting the previous data (D₀) up to the total number of 1’s present in that data (SD₀) XOR with right shifting the previous key (K₀) up to the total number of 1’s present in that key (SK₀). So the new key will be K₁= SD₀ XOR SK₀ = 00010100 XOR 10101010 = 10101110.

**C. ILLUSTRATION OF DSAVK**

Let sender sends initial data D₀ (00000000) in encrypted form using key K₀(00000110). By the technique of DSAVK as in eqn.(3), the next key will be generated by right shift operation and will be K₀=00011000 (Right shift will be up to decimal equivalent of (D₀ XOR K₀)).

**V. ANALYSIS AND COMPARISON**

For analysis and comparison of Protocol-I, Protocol-II, Protocol-III we assume a parameter of randomness as a measure of amount of variation made between the successive keys. The randomness for the purpose is defined as the number of bit location in which any two successive key vary. For example if:

\[ K_{i}=10101010, \quad K_{i+1}=10011111. \]

The randomness between two successive key is 3. We call Kᵢ₊₁ is random to Kᵢ by 3.

For the three new technique the set of initial keys are \{11001010, 10101100, 11111111\}, initial numeric numbers are \{16, 16, 16\} and perform key generation for the set of initial data \{00000000, 00000001, 00000010, 00000011, 00000100, 00000101, 00000110, 00000111, 00001000, 00001001, 0100, 00001111, 00011000, 00011011, 00011110, 00011111, 00000000, 0000001, 00000010, 00000011, 00000100, 00000101, 00000110, 00000111, 00001000, 00001001, 00001010, 00001011, 00001100, 00001101, 00001110, 00001111, 00010000, 00010001, 00010010, 00010011, 00010100, 00010101, 00010110, 00010111, 00011000, 00011001, 00011010, 00011011, 00011100, 00011101, 00011110, 00011111, 00000000, 00000001, 00000010, 00000011, 00000100, 00000101, 00000110, 00000111, 00001000, 00001001, 00001010, 00001011, 00001100, 00001101, 00001110, 00001111, 00010000, 00010001, 00010010, 00010011, 00010100, 00010101, 00010110, 00010111, 00011000, 00011001, 00011010, 00011011, 00011100, 00011101, 00011110, 00011111, 00100000, 00100010, 00100011, 00100100, 00100101, 00100110, 00100111, 00101000, 00101001, 00101010, 00101011, 00101100, 00101101, 00101110, 00101111, 00110000, 00110001, 00110010, 00110011, 00110100, 00110101, 00110110, 00110111, 00111000, 00111001, 00111010, 00111011, 00111100, 00111101, 00111110, 00111111\}.

The randomness as defined was calculated by run of a programme and results so obtained are portrayed in fig.1, fig.2 and fig.3.

**VII. CONCLUSIONS**

From the comparison of the results portrayed in fig. (1-3) it is found that:

- **Randomness for the set of data and set of initial key under experiment, is more in protocol-II and in protocol-III than that in protocol-I**
- **Randomness as measure under the same experiment is more in protocol-II than that in protocol-III.**
- **In term of randomness in variant key, protocol-II is superior to both protocol-III & protocol-I.**

We propose to apply the techniques in AES; and examine brute force attack & differential frequency attack in subsequent studies.

**REFERENCES**


Rajat Subhra Goswami, Swarnendu Kumar Chakraborty, Abhinandan Bhunia, C. T. Bhunia, “Generation of Automatic Variable Key under various approaches in Cryptography System”, Communicated to Journal of the Institution of Engineers (India).

fig.1: Randomness of keys of Protocol-I

fig.2: Randomness of keys of Protocol-II

fig.3: Randomness of keys of Protocol-III