A novel approach towards realizing optimum data transfer and Automatic Variable Key(AVK) in cryptography

P. Chakrabarti[1], B Bhuyan[2], A.Chowdhuri[3], C.T.Bhunia[4]
[1]Dr. B.C.Roy Engineering College, Durgapur-713206, India
[2] Haldia Institute of Technology, HIT-721657, India
[3]Jadavpur University, Kolkata-700032, India
[4] Haldia Institute of Technology, HIT-721657, India and ICTP, Trieste, Italy

Summary
High Level Information security needs research and Investigation due to increasing security threats & attacks with increasing volume of Information traffic. In literature huge studies are made but all with a fixed secret key, but variable encryption scheme and key transport protocol. Hardly any research has attempted to address the issue of time variant key, where the secret key will vary from session to session. Shannon documented the theory of perfect secrecy with time variant key. An idea of time variant key in name of Automatic Variable Key has been recently introduced [3-5]. This paper deals with the efficient transmission of data from source to destination dynamically based on optimum path selection in certain distributed models[6]. The concept behind this security enhancement is that a shared secret key is agreed upon for a session. In order to solve key distribution problem, use of quantum channel[7] for sending information about key is being explored. A single photon can represent a bit 0 or 1. The phase or state of polarization of the photon may be used for identifying the 0 or 1. In this paper it has been shown that security enhancement through quantum channels can be ensured by varying the key, that is, changing the phase using non-orthogonal measurement bases It has also been shown how AVK can be applied in Vernum Cipher. This paper also points out application of AVK (Automatic Variable Key) in curves[8,9], AES[10] , RSA[11,12] , diffusion and message digest [13]and it is shown that it increases security level during transmission.

Key words:
optimum path selection, automatic variable key , quantum cryptography, AES, RSA , diffusion , message digest

1. Introduction

In all cryptosystem the challenge of the designer is to make key unbreakable whereas the challenger threats to break the key. Vernum 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. Recently AVK was proposed as a time variant key. The proposed AVK is illustrated in the table below for a session between Alice and Bob whereby they respectively exchange data 34 and 78. In AVK, the key is made variable with data.

K_0 = initial secret data, $K_i = K_{i-1}$ XOR D_i for all i>0 where D_i = data in ith session

TABLE1: Illustration of application of simple AVK in cryptology

Session slots	**Alice sends**	**Bob receives**	**Bob sends**	**Alice receives**	**Remarks**
1	secret key (say 2)	2	secret key (say6)	6	For next slot, Alice will use 6 as key and Bob 2 as key for transmitting data
2	Alice sends first data as 3XOR6	Bob gets back original data as (3XOR6XOR6) = 3	Bob sends first data as 7XOR 2	Alice gets back original data as (7XOR2 XOR2) = 7	Alice will create new key (6XOR7) for next slot. Bob will create new key (2XOR3)
3	Alice sends next data as 4XOR6 XOR 7	Bob gets back original data as (4XOR6XOR 7XOR 6XOR 7) = 4	Bob sends next data as 8XOR 2XOR 3	Alice recovers data as (8XOR2 XOR3 XOR2 XOR3) = 8	Thus Alice and Bob respectively exchange data 34 and 78

Manuscript received May 5, 2008
Manuscript revised May 20, 2008

2. Communication in cube

To find out the number of channels required for communication from a processor to its farthest processors in a distributed system where there are eight processors placed in a cubicle fashion where the nodes are communicating processors.

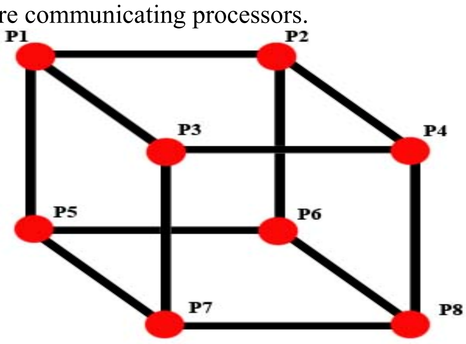

Fig 1. Cubic Model

Since the model is of distributive nature the sharing of information between receiver and sender should take place in an optimum way.

As per the model the nodes denote the processors. Let us assume that P1 is the source processor. We require the communication to be between the source processor and its farthest processor in an optimum way i.e. the number of channels or paths are necessary and sufficient. Hence the farthest processor from P1 (the source processor) in this case is P8.

Thus, the number of communication channels = 3.

The possible paths in which optimized communication may take place are:

P1 → P2, P2 → P4, P4 → P8 or,
P1 → P2, P2 → P6, P6 → P8 or,
P1 → P3, P3 → P4, P4 → P8 or,
P1 → P3, P3 → P7, P7 → P8 or,
P1 → P5, P5 → P7, P7 → P8 or,
P1 → P5, P5 → P6, P6 → P8.

Optimum transmission means that the number of channels will be based on a certain criterion such that the cost complexity as the time required are minimized provided there are no transmission errors or interference of any noise (spikes) .

Thus, we may mathematically denote the number of channels required for communication between the source and its farthest processor placed in a cubic model as $\log_2 N$, where N= total number of processors.

3. Communication in hyper-cubic model

In the case of the processors being arranged in a hyper-cubical fashion the total number of processors are 16. Now the number of channels required for a processor to communicate with its farthest processor will be one more than the number of communication channels if there were only one cube. Thus the number of communication channels = 4.

Thus, the general formula $\log_2 N$, where

N= total number of processors is applicable to a hyper-cubical model.

4. Optimum communication arranged in matrix model

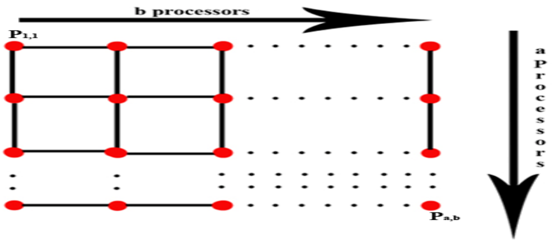

Fig 2: Matrix model of the form a X b

In this model we consider $P_{1,1}$ to be the source processor. Thus the farthest processor is $P_{a,b.}$ Now, for $P_{1,1}$ to communicate with $P_{a,b}$, (b-1) channels must be covered horizontally and (a-1) channels must be covered vertically. Thus the total number of channels required for communication of the source processor to the farthest processors can be mathematically stated as (a-1) + (b-1).

To support the above logical deduction let us consider a mathematical example.

Let a=2 and b=3. Thus the arrangement is as follows:

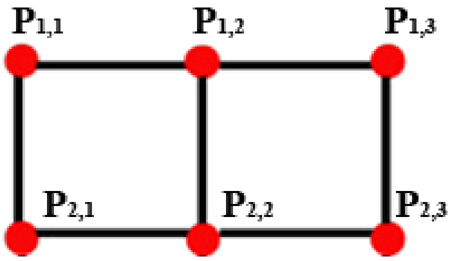

Fig 3: A sub model

$P_{1,1}$ communicates with $P_{2,3}$ by taking one of the following paths:

$P_{1,1} \rightarrow P_{1,2}$; $P_{1,2} \rightarrow P_{1,3}$; $P_{1,3} \rightarrow P_{2,3}$ or,
$P_{1,1} \rightarrow P_{1,2}$; $P_{1,2} \rightarrow P_{2,2}$; $P_{2,2} \rightarrow P_{2,3}$ or,
$P_{1,1} \rightarrow P_{2,1}$; $P_{2,1} \rightarrow P_{2,2}$; $P_{2,2} \rightarrow P_{2,3}$.

Thus we see for optimized communication to take place, 3 channels need to be covered which satisfies the above deduced formula (a-1) + (b-1) [(2-1) + (3-1) = 3].

To find out the general formula for the number of communication channels required for optimal communication between a processor and its farthest processor. The processors being arranged in a X a square matrix model.

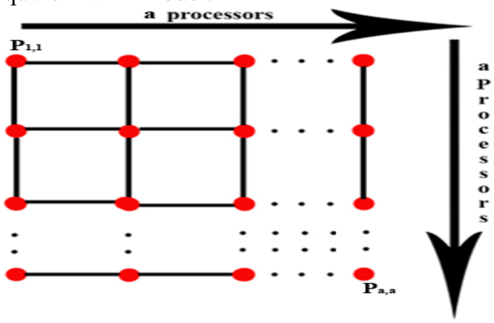

Fig 4 Matrix model of the form a X b

In the previous model if we replace b=a we get a square matrix of the form a X a.
We know that for a matrix of a X b form,
Number of communication channels required for optimized communication
= (a-1) + (b-1),Replacing b = a, we get,
Number of communication channels required for optimized communication
= (a-1) + (a-1) = 2(a-1)
To support the above deduction let us consider a mathematical example.Let a=3. Thus the arrangement is as follows:

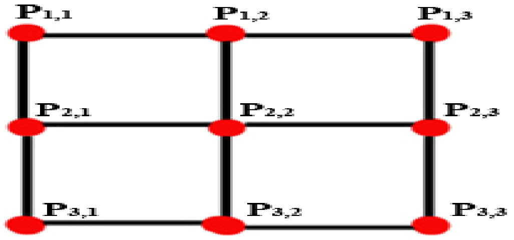

Fig 5: Illustration of Matrix model of the form 3 X 3

$P_{1,1}$ communicates with $P_{3,3}$ by taking one of the following paths:
$P_{1,1} \rightarrow P_{1,2}$; $P_{1,2} \rightarrow P_{1,3}$; $P_{1,3} \rightarrow P_{2,3}$; $P_{2,3} \rightarrow P_{3,3}$ or,
$P_{1,1} \rightarrow P_{1,2}$; $P_{1,2} \rightarrow P_{2,2}$; $P_{2,2} \rightarrow P_{2,3}$; $P_{2,3} \rightarrow P_{3,3}$ or,
$P_{1,1} \rightarrow P_{1,2}$; $P_{1,2} \rightarrow P_{2,2}$; $P_{2,2} \rightarrow P_{3,2}$; $P_{3,2} \rightarrow P_{3,3}$ or,
$P_{1,1} \rightarrow P_{2,1}$; $P_{2,1} \rightarrow P_{2,2}$; $P_{2,2} \rightarrow P_{2,3}$; $P_{2,3} \rightarrow P_{3,3}$ or,
$P_{1,1} \rightarrow P_{2,1}$; $P_{2,1} \rightarrow P_{2,2}$; $P_{2,2} \rightarrow P_{3,2}$; $P_{3,2} \rightarrow P_{3,3}$ or,
$P_{1,1} \rightarrow P_{2,1}$; $P_{2,1} \rightarrow P_{3,1}$; $P_{3,1} \rightarrow P_{3,2}$; $P_{3,2} \rightarrow P_{3,3}$.
Thus we see for optimized communication to take place, 4 channels need to be covered which satisfies the above deduced formula 2(a-1) [2(3-1) = 4]

5. AVK in quantum technology

The disadvantage of key distribution can be removed with the aid of quantum technology. If key distribution problem is solved, the use of Vernum technique will be best technique of security. In order to solve distribution problem, use of quantum channel for sending information about key is being explored. In quantum mechanics one cannot measure something without causing noise to other related parameter. For example, Hysenberg's uncertainty principle states that x.m = constant. Thus if Δx is changed, Δm is bound to change. An ideal quantum channel supports transportation of thr single photon. Thus a single photon can represent a bit-0 (zero) or 1(one). Sender will transmit key and receiver will sense photon bits and the bit values taken together will constitute common key. For enhancing speed, data can be sent from sender to receiver through multiple quantum channels.

5.1.Method 1

5.1.1.Principle

(1) Alice sends secret key required in Vernum cipher for Bob through
quantum channel. Phase is same throughout transmission of entire
data.
(2) Alice instructs Bob to detect the photon (bits) from the quantum
channel starting from a given time.
(3) Inspite of transmission loss, Bob may be able to detect some
fraction of photons.
(4) Bob informs Alice as to which photon he has seen.
(5) Hence they share a common key.

5.1.2. Numerical analysis

Let Alice sends 11010010 and Bob replies that he has seen the first, seventh and eighth photons (starting from leftmost bit). Then the common key will be 110.

.

5.1.3. Time complexity

Let us suppose that n = number of bits in the key. Bob has to sense each and every bit of the key. If transmission time of a single bit from Alice to Bob is t_1, then time to sense 1^{st} bit is t_1. Then after each delay Bob will sense next one.So, total time taken = (t_1+n-1)It is assumed that time to sense a bit is negligible and pipeline has been taken into consideration.

5.1.4.Disadvantage

Since the phase is same throughout of entire message transmission, so if a hacker hacks the key, then since the phase is same, so there is a clear-cut idea of him to sense the photon bits and hence Bob will get erroneous value.

5.2.Method 2

5.2.1. Principle

(1) Alice encrypts the data using its private key K_A.

(2) Alice sends the encrypted form to Bob.
(3) Bob decrypts using same key K_A.
(4) Bob senses the bits and passes the information.

In this case also, phase of photon is same throughout transmission of entire message.

5.2.2. Numerical analysis

Let Alice sends 11010010 in encrypted form using key 100. Then in order to maintain linearity, the encrypted form is 11010010 XOR100 = 11010110

At receiver end, Bob will perform 11010110 XOR 100 and get 11010010. Then it will sense the photon bits.

5.2.3.Time complexity

Time taken = t_e+ (t_l+n-1) +t_d+nt_s

where t_e = time for encryption

t_l = time for transmission of a single bit from Alice to Bob.

n = total number of bits.

t_d = time for decryption

t_s = time for sensing photon bit at a particular index position

5.2.4.Disadvantage

Since the private key k_A is fixed, so in case of multiple transition if k_A is hacked then information will be leaked from next session. Another disadvantage is that an intruder could interrupt the signal, copy Alice's message and send it on to Bob without either Alice or Bob realizing.

5.2.5.Handling attacks

Eavesdropping can be tackled by sending photons with different phases. For example, the 0 may be represented by a photon having a phase of 0° or 180°, and the bit 1 may be denoted by a photon with a 90° or 270° phase. When Bob uses, he will be able to detect the bits correctly.

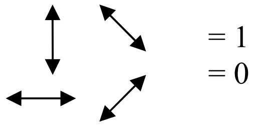

Fig. 6 : Use of polarization for representing 1s and 0s

Alice haphazardly uses both to send qubits (fig 2). Bob will haphazardly try to filter out the qubits. For the purpose of qubits. For the purpose of qubits detection Bob will use a polarization beam splitter. The polarized beam splitter is a device that allows the photons of orthogonal polarization to pass through but shunts the photon of other polarization. The quantum nature dictates that (a) the same basis beam splitter will pass the received same basis polarized photons, but (b) the rectilinear beam splitter will pass the received diagonally polarized photons either as vertical or horizontal polarization will equal probability and the diagonal beam splitter will pass the received rectilinear polarized photons either as vertical or horizontal polarized photons with equal probability.

↕ rectilinear	⤢ diagonal	↔ rectilinear	↕ rectilinear	⤡ diagonal

Fig. 7 : Alice sends quits to Bob randomly (we have taken 5 quits)

↔ rectilinear	↔ diagonal	⤢ rectilinear	↔ rectilinear	⤢ diagonal

Fig. 8 : Bob measures the received photons using random polarization basis

In case of multiple transmission, Alice can send the data haphazardly using different polarized photons. Alice can do so either on rectilinear basis or on a diagonal basis. When a horizontal polarized photon represent a 0 and a vertical polarization represent a 1. When a -450 polarized photon represent a 0 on diagonal basis, then a +450 polarized photon represent a 1.

↕ Correctly Detected by Bob	Uncertain	Uncertain	↕ Correctly Detected by Bob	⤡ Correctly Detected by Bob

Fig9: Alice and Bob communicate and identify locations whether they correctly used the polarization base. Alice and Bob keep secret the polarization of sent or received photons.

5.3. Method 3

Session slots	Alice sends	Bob receives	Bob sends	Alice receives	Remarks
1	secret key K_A (say 2)	2	secret key K_B (say 6)	6	For next slot, Alice will use 6 as key and Bob 2 as key for transmitting data
2	Alice sends first data as 3XOR6	Bob gets back original data as (3XOR6XOR6) = 3	Bob sends first data as 7XOR2	Alice gets back original data as (7XOR2 XOR 2) = 7	Alice will create new key (6XOR7) for next slot. Bob will create new key (2XOR3)
3	Alice sends next data as 4XOR6 XOR 7	Bob gets back original data as (4XOR6XOR 7XOR6XOR7) = 4	Bob sends next data as 8XOR2XOR3	Alice recovers data as (8XOR2 XOR 3XOR2 XOR 3) = 8	Thus Alice and Bob respectively exchange data 34 and 78

Fig10: Key exchange between Alice and Bob

5.3.1. Principle

(1) Alice encrypts the data using its private key K_A

(2) Alice sends the encrypted form to Bob.

(3) Bob decrypts using same key K_A.

(4) Bob senses the bits and passes the information.

(5) Time variant key is applied, i..e. K_A is subject to change from session to session.

5.4. Method 4

Principle: An eavesdropper could intercept the signal, copy Alice's message and send it on to Bob without either Alice or Bob realizing. One way to overcome this, and ensure absolute security is for transmitter and receiver to use non-orthogonal measurement bases. In other words, Alice sends parts of the message by switching the transmitter phase between 90° and 270° say, and other part by switching between 0° and 180°. When Bob and Alice using same base, the system works as before. However if Alice using 0°/180° and Bob is using 90°/270° (or vice-versa), the message is meaningless. Hence a photon that Alice sends as a "0" has a 50% chance of being received as a "1" and vice-versa. Therefore when Bob tells Alice which base he was using and Alice must tell him if it is a valid photon or not.

5.5. Method 5

5.5.1. Principle

Another technique to minimize the hacking by privacy amplification protocol. In the protocol, Alice randomly choses pair of bits from the key they have got over quantum channel. Then she performs XOR on the pairs. She then tells publicly to Bob on which bits the XOR operation was made but not the results. Bob then performs the XOR operation on the bits that Alice informed him. Alice and Bob then replace the pair with XOR results to design the new key.

5.5.2. Numerical analysis

The steps are as below:

(a)Alice and Bob have secret key 111

(b)Alice chooses first and second bit as pair and she informs these to Bob publicly. She gets XOR result 1+1 = 0 and keeps it secret.

(c)Bob performs XOR on the informed bits and get the result 1+1 = 0.

(d)Alice and Bob both replace the pair by XOR result. So their new key = 01.

(e)Even if hacker definitely knows one bit of the chosen pair, until and unless she gets the result of XOR (which Alice and Bob never communicates) she cannot replace the pair of for hacking the key.

6. Vernum cipher using AVK

6.1. Algorithm

1. Assume a secret key (k) of n bits
2. Break the message into blocks each of n bits. Let block are m_1, m_2, …..m_p
3. generate encrypted blocks as ci= mi XOR k , for i=l to p. Then transmit the encrypted blocks.
4. Receiver will decrypt the blocks as y_i=c_i XOR k for i=l to p. Note that y_i= c_iXORk=m_i.

All the blocks received under deciphering will constitute the plaintext.

6.2.Enhancing security level

Vernum variable key is to be introduced. Consider p number of sessions. Each session takes a time of t seconds for completion on average. Assume key size of n bits. Apply bite force attacks for getting the session keys. Under non-variables /one time vernum key,1he hacker may try on average 2^{n-i} trials over a period of pt seconds. The required time for analysis of a pattern will be than $(pt/2^{n-i})$ seconds. Under vernum variable key The hacker has to try on the average 2^{n-i}trial over a period of single session (t seconds). This is because the key will change from session to session. Thus the required time of analysis will be $(t/2^{n-i})$ Now, as $(pt/2^{n-i})>(t/2^{n-i})$, the attack is more effective in the one time key by an order of p. hence vernum code is more secured over one time key by an order of p. Ideal case: if p=1, then both cases are same.

7. Cipher generation based on feature of straight line

Let Fibonacci series be 1, 1, 2, 3, 5, 8, 13, 21. Sender has x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, i.e. integers .For straight line, y=a+bx

Step1 : R_1= a + b* d_1 = 1 + 1*d_1 , where a=1=first term of Fibonacci series, b=1=second term, d_1=first data. Therefore, cipher 1= R_1XOR k_1 , where k_1= key for first session =x_1XORy_1

Step2 : R_2= 2+3*d_2.

Cipher2= R_2XORK_3=R_2XORx_2XORy_2

Step3: R_3=5+8*d_3.

Cipher3= R_3XORK_3= R_3XORx_3XORy_3

Step4: R_4= 13+21*d_4.

Cipher4= R_4XORK_4= R_4XORx_4XORy_4

If hacker knows random number sequence and ciphers of each stage, then by calculation it can get the data of each stage. Suppose, hacker knows random sequence, a and b, cipher 1.Therefore, cipher q=R_1XORK_1.Now initially for y=a+ bx , it can get k_1, k_2, k_3, k_4So he can get R_1.Now R_1= a+b*d_1. So he can hack d_1 Similarly, d_2, d_3, d_4 will be hacked. Solution is d_2= $\log_{d1}k_2$ where k_2=x_2XORy_2XORd_1.

8. Cipher generation based on features of parabola, Gompertz curve

a=1, b=1, c=2, x=d. Therefore, y_1= a + b* d_1 + c*x^2 = 1+1*d_1+2*d_1^2 =1+d_1+2d_1^2

Therefore, Cipher 1= y_1XOR[$\log_d$e] where d= 4th term is series=3 , e= 5th term in series=5

So cipher of previous stage depends on the random number of next stage[3][4].If hacker knows random sequence and cipher1, it can get d_1So, solution is Cipher2= y_1XOR[$\log_d$(en)] , where n is a number private to sender only.The equation of Gompertx curve is log y= a+bc^x. R1= log(a+bc^x). Cipher1= R_1XOR[$\log_d$e].For enhancing security, cipher1 = R_1XOR[$\log_d$en]. Even if hacker knows R_1, difficult to find x as inverse logarithmic function takes much time and it is infeasible.

9. Cipher generation based on feature of imaginary variable

9.1 Approach 1

Input : d_1, k_1. Now R_1= d_1+ik_1.Cipher1= R_1*(d_1-ik_1)= (d_1+ik_1) (d_1-ik_1)= d_1^2+k_1^2

If hacker hacks k_1 and cipher1, it will get d_1. Solution is : R_1=$(d_1-ik_1)^2$ =$(d_1^2-k_1^2)$ – $2d_1k_1i$

Cipher1= | (coefficient of real) | XOR |Co-efficient of imaginary|= | $(d_1^2-k_1^2)$ | XOR $(2d_1k_1)$.This extraction scheme of extraction of co-efficients of real and imaginary parts are known to user only. Advanced security level is applied. k_2 of next step= d_2XORk_1. Therefore ,

R_2 = $(d_2-ik_2)^2$ = $(d_2^2-k_2^2)$ – $2d_2k_2$. Cipher2= | d_2^2 – k_2^2| XOR $(2d_2k_2)$

9.2 . Approach 2

Sender choose random numbers a and b **.**R_1= $(d_1 + iak_1)^2$ = d_1^2+$i_2a_2k_1^2$+$2d_1iak_1$ = $(d_1^2 - a^2k_1^2)$ + $(2d_1ak_1)$i**.** Thus Cipher1= $(d_1^2 - a^2k_1^2)$ XOR $(2d_1ak_1)$**.**From next step onwards, k_2 = $\log_c$d XOR k_1 where c,d are random numbers for next session.

10. Cipher generation based on AVK in AES

10.1. Concept

Using normal AES the encryption is done. Now if there is redundancy of data block then that will affect the cipher. So selective AES will be taken into consideration. But it also has a demerit as the key is constant in each session. So AVK with selective AES is done.

10.2. Numerical analysis

Plain text:

Secret key **FEG4** Secret key **FEG4** is changed to **7A**

Normal AES

Cipher :

a8 48 60 e a6 d 10 0 18 4e fa 1 56 5d 6e 34

a8 48 60 e a6 d 10 0 18 4e fa 1 56 5d 6e 34

9a 1d ed 14 c3 64 47 e1 59 41 79 13 17 19
cb d4

Key: 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf
4f 3c

Hence there is repetition in cipher(in bold) and reduces security level.

Selective AES

Cipher :

6d dc a5 b ea 24 fe c3 d3 72 aa a7 46 eb 7e 98
6d dc a5 b ea 24 fe c3 d3 72 aa a7 46 eb 7e 98
70 39 d0 26 59 a5 6 d2 30 21 e 1e f2 38 e7 61

Key : 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf
4f 3c

Using selective AES with AVK

Cipher :

67 e3 59 c9 83 99 fd 9e c9 11 60 f1 e c7
70 d2
73 32 37 3d c0 f4 69 4f df 7d ee 5f a7 63
25 d7
ed 76 44 c 9a bb db 50 57 7c 3f 58 8c e3
7c ca

Key1**:** 2b 7e 15 16 28 ae d2 a6 ab f7 15 88
09 cf 4f 3c
Key2**:** 4c 9d 4c df ab 37 2f 3c 62 e6 75 79 7
8 3f ee
Key3**:** 3f af 7b e2 6b c3 46 73 bd 9b 9b 26 a0
6b 1a 39

11. Cipher generation based on AVK in RSA

11.1. RSA with AVK method 1

The encryption of the algorithm is as follows:

1) The RSA keys will be generated as per normal RSA algorithm.

2) Each character of the input data is converted into integer by substituting its respective ASCII value.

3) Encrypt first byte i.e. A_0 and send it as it is.

4) Send encrypted (A_0xorA_1) as the next byte.

5) The nth byte is encrypted as Encrypted (A_0xorA_1xor…A_n).

The decryption of the algorithm is as follows:

1) Decrypt the entire string.

2) The first byte is exactly the same.

3) The second byte is E_0xorE_1.

4) The next byte is xor of it and the previous one.

5) The nth byte is xor of it and its previous byte.

6) Convert the ASCII bit stream back into the message.

11.2. RSA with AVK method 2

The encryption of the algorithm is as follows:

1) The RSA keys will be generated as per normal RSA algorithm.

2) Each character of the input data is converted into integer by substituting its respective ASCII value.

3) Encrypt first byte and send it as it is.

4) The next byte is sent as Encrypted (A_0xorEncrypt(A_1)).

5) The nth byte is encrypted as Encrypted (A_0xorA_1xorA_2….xorEncrypted(A_n)).

The decryption of the algorithm is as follows:

1) Decrypt the message using RSA decryption algorithm.

2) The first byte is available as it is.

3) Next byte can be obtained by xor operation of first byte with it and then decrypting the entire value.

4) The nth byte is obtained by xor operation of first (n-1) bytes with nth byte in decrypted form and at last decrypting the entire value.

5) Convert the ASCII bit stream back into the message stream.

11.3. Analysis of RSA with AVK

Plain text

A message is encrypted. Key is 1101. Another message is encrypted. Key is 1101.

Results using Normal RSA

8e 4c 83 54 9d 9d 5c 89 54 4c 60 9d 4c
54 42 b0 7e 4d 49 4a 54 90 7 4c 72 **54**
4d 4c 60 9d 4c 19 19 9f 19 7 8e 42 9b
4a b3 54 7e 4c 83 54 9d 9d 5c 89 54 4c
60 9d 4c 54 42 b0 7e 4d 49 4a 54 90 7
72 **54 4d 4c 60 9d 4c 19 19 9f 19 7** af

User time + System time = 0.001 + 0.001 = **0.002**

Results using RSA with AVK (Method 1)

8e 51 24 8e 6e a0 b8 9 30 87 11 15 1a
97 69 af 22 1f 9b 23 3c b b5 8c b1 b9
92 75 ad 12 76 71 4 23 ab 17 67 46 2e
5f 84 2f a8 aa b1 b9 5d 7 77 87 81 53
61 ab 1c 4d 12 14 77 6c ad 7b 7b 7d 8
43 2f 67 71 1d 42 15 9 4e 50 5c 7e 4a

User time + System time = 0.001 + 0.002 = 0.003

Results using RSA with AVK (Method 2)

```
8e b6 1a 38 8f ab 24 4f 7c 9f 33 51 a0
96 b9 46 4e 82 87 ab 33 9e 99 68 68 24
60 16 65 6b 81 a9 63 8e b2 3 b4 31 66
16 8f 4c 76 61 81 68 61 8e 85 b1 f 43
12 a5 7 8 a7 85 4 3d 4a 0 32 2 92 b5
2f 15 a6 58 af 74 83 9a 92 1c 7c 1d
```

User time + System time = 0.002 + 0.003 = 0.005

11.4. Different cryptanalysis attacks on the proposed schemes

Consider two messages M1 and M2.
M1: The price of copper is 100.
M2: The price of copper is 200.
In either scheme this can be written as :
A0 a0 a1 a2 and A0 a0’ a1’ a2’
where A0 is the byte before the numerical value of the price of copper. Since each message has the same string before the numerical value A0 is the same. The messages are encrypted as :
Message1 : E(A0) E(A0 XOR a0) E(A0 XOR a0 XOR a1) E(A0 XOR a0 XOR a1XOR a2)
Message2 : E(A0) E(A0 XOR a0’) E(A0 XOR a0’ XOR a1’) E(A0 XOR a0’ XOR a1’XOR a2’)
Now the attacker can guess that E(A0) can stand for the string “The price of copper is”. He then intercepts the message and changes it to as follows :
E(A0) E(A0 XOR a0 XOR a1) E(A0 XOR a0) E(A0 XOR a0 XOR a1XOR a2)
Decrypted, this becomes A0 , A0 XOR a0 XOR a1 , A0 XOR a0 , A0 XOR a0 XOR a1XOR a2
which when decrypted becomes A0 a0 XOR a1 a1 a1 XOR a2
Now, depending on the contents of the message , a0 a1 a2 this may or may not make sense.
For example if a0 a1 a2 = 332 , then it is received as 031
Hence an attacker is able to change the contents of a message in the method 1 of RSA with AVK.
In the method 2 of RSA with AVK , this becomes
A0 A0 XOR E(a0) A0 XOR a0 XOR E(a1) A0 XOR a0 XOR a1 XOR E(a2)
Thus if the order is mixed to get :
A0 A0XOR a0 XOR E(a1) A0 XOR E(a0) A0 XOR a0 XOR a1 XOR E(a2)
The receiver will receive this as
A0 E’(a0 XOR E(a1)) E’(a0 XOR E(a1)XORE(a0)) E’(E(a0) XOR a0 XOR a1 XOR E(a2))
where E’ = Inverse RSA = decryption. Thus in general it will not make sense to the receiver .
Clearly , here method 1 of RSA with AVK fails while method 2 works.

12. Application of AVK in diffusion

12.1. Concept

The rule that has been proposed is that the key of each session will be determined based on the data of that session. The formula is as follows:

$$x \log_2 d/k = 1$$

where x = session number
d = data in that session
k = key of that session
After finding the key , encryption is done with the help of any standard system.

12.2. Numerical analysis

Data1 : 5 ; Data2 : 6 ; Data3 : 8

Step1: x = 1 , d1 = 5
Therefore 1 .* $\log_2$ 5/ k1 = 1
or , $\log_2$ 5/ k1 = $\log_2$ 2
or , 5 / k1 = 2
or, k1 = 3 (taking upper ceiling)

Step2 : x = 2 , d2 = 6
Therefore 2 .* $\log_2$ 6/ k2 = 1
or , $\log_2$ 6/ k2 = 0.5
or , $2^{0.5}$ = 6 / k2
or , 1.41 = 6 / k2
or, k2 = 5 (taking upper ceiling)

Step3: x = 3 , d3 = 8
Therefore 3 .* $\log_2$ 8/ k3 = 1
or , $2^{1/3}$ = 8/ k3
or , 1.25 = 8 / k3
or, k3 = 7 (taking upper ceiling)
It is seen that the keys are as follows:

SESSION NUMBER	KEY
1	3
2	5
3	7

So the key range is in A.P. with interval 2 and
the formula is KEY = 2 * SESSION NUMBER + 1

12.3. Application of AVK

To overcome above limitation , AVK is applied where the first key is calculated as above and from second session onwards
KEY_FINAL = (KEY_CALCULATED) XOR (PREVIOUS CIPHER**)**
Data1 : 5 ; Data2 : 6 ; Data3 : 8 and the Encryption technique is simple XOR.

Step1: x = 1 , d1 = 5
Therefore 1 .* $\log_2 5/ k1 = 1$
or , $\log_2 5/ k1 = \log_2 2$
or , 5 / k1 = 2
or, k1 = 3 (taking upper ceiling)
So cipher1 = k1 XOR Data1 = 3 XOR 5 = 6

Step2 : x = 2 , d2 = 6
Therefore 2 .* $\log_2 6/ k2 = 1$
or , $\log_2 6/ k2 = 0.5$
or , $2^{0.5} = 6 / k2$
or , 1.41 = 6 / k2
or, k2 = 5 (taking upper ceiling)
So effective key of session 2 = k2 XOR cipher1 = 5 XOR 6 = 3
So cipher2 = 3 XOR Data2 = 3 XOR 6 = 5

Step3: x = 3 , d3 = 8
Therefore 3 .* $\log_2 8/ k3 = 1$
or , $2^{1/3} = 8/ k3$
or , 1.25 = 8 / k3
or, k3 = 7 (taking upper ceiling)
So effective key of session 3 = k3 XOR cipher2 = 7 XOR 5 = 2
So cipher3 = 2 XOR Data3 = 2 XOR 8 = A_{16}

It is seen that the keys are as follows:

SESSION NUMBER	KEY	CIPHER
1	3	6
2	3	5
3	2	A

So no proper relation between any pair .

13. Application of AVK in message digest

13.1. Encryption scheme

We propose a new AVK scheme .It is as follows:

1. Calculate the message digest C
2. Now take a security parameter Y where Y is the number of bytes necessary for encoding one byte. Y should be a power of 2.
3. Make a0 as the first byte and C as the last byte of an array of 5Y digits: [a0 0 0 C], Y is say 4
4. Calculate the discrete Fourier transform of the function.
5. The first Y bytes are then the encrypted version of this digit.
6. The same process is repeated . Except that C is put in a position a0 mod (Y-1)+1 position. This shifting of C is done to further increase security. Instead of putting the first byte as a1 , Hash(a0)XORE(a1) is pit.
7. Thus the nth byte is encrypted into Y bytes , where the bytes are the DFT of an array with the first byte being [Hash(a0) XOR Hash(a1) XOR…..Hash(a_{n-1}) XOR E(a_n) 0….0…C] ,where the position of C is dictated by the previous byte , as said before.

13.2. Decryption scheme

1. First the entire message stream is decrypted.

2. The first Y bytes are taken , and their Inverse DFT is found out. The first byte is a0. (This can be obtained by passing the data through a low pass filter and then getting the first byte or directly).

3. The next 5 bytes ,IDFT is found. Then using the first byte, a filter is constructed to get the second byte which is Hash(a0) XOR E(a1). This is then extracted .

4. This is continued for all (n-1) bytes.

13.3. Time complexity

In general if N is the size of the plain text , then the size of the cipher is N*Y where Y as discussed before is the security parameter. The speed of encryption is considerably slower and is equal to N*Y*(T(E)+T(F)) , where T(E) = time required for the encryption of the data , T(F) = time required to calculate the Fourier transform. The speed of decryption is similar. The disadvantage is the increase in cipher text size. Hence it is much slower and requires huge amount of bandwidth. A solution to this problem is to compress the data after calculating the DFT. Another solution is to use other transforms like the Walsh Hadamard Transform , the Gilbert’s Transform or the Discrete Cosine Transform .

14. Conclusion

The paper reveals that data communication can be done based on optimum path selection. Gain in time complexity can be achieved by parallel computing and parallel reliability. The paper hints several methods regarding transmission of data (photons) through quantum channel. It has been seen that the most secured criteria is variable data with variable secret key with message transmission mode also being variable (i.e. both transmitter and receiver to use non-orthogonal measurement bases). It has also been shown how AVK can be applied in Vernum Cipher. Also for security purpose of data transfer between nodes , several intelligent and efficient schemes of security can be applied based on basic properties of straight line, parabola, Gompertz curve and imaginary variable. The paper shows superiority of application of AVK in AES, RSA , diffusion and message digest over conventional schemes.

References

[1] C T Bhunia, IT, Networks and Internet, New Age Int'l Publishers, New Delhi, 2005

[2] Martin E Hellman, An Overview of Public Ket Cryptography, IEEE Communication Mag, May'2002, PP. 42-49

[3] W Diffe and M E Hellman, Exhaustive Cryotanalysis of the NBS Data encryption standard, Computer, June'1977, PP. 74-84.

[4] E Biham, A fast new DES Implementation in Software, Proc. International Symp. Foundations of Software Engineering, FSE'97, PP.260-273

[5] H Eberle, A High Speed DES Implementation for Network application, Proc. International Conf. Cryptology, CRYPTO'92, 12993, PP.521-539

[6] P.Chakrabarti , D. Banerjee, Optimum data transfer and related security", published in National Conference on Methods and models in Computing, Jawaharlal Nehru University, India, December 2007

[7] C.T.Bhunia , P.Chakrabarti , B.Bhuyan , A.Chowdhuri "Enhancement of security level in quantum computing and that of speed by parallelism", published in proceedings of INTL-INFOTECH 2007, HIT , Haldia , March'07, pp749-752

[8] P.Chakrabarti "Intelligent schemes of cipher generation using comparison analysis and curves" ,International Conference on IT , Jabalpur, Dec07,pp 299-301

[9] B.Bhuyan , P.Chakrabarti , S R Bhadra Chowdhury , C.T.Bhunia ,"Experimental Studies on Different Approaches of implementing AVK, Time Variant Key on Information Security",selected and to be published in IEEE CIT08, Australia, July2008

[10] P.Chakrabarti , G.H.Mondal , C.T.Bhunia ,A.Chowdhuri , "Various New and Modified Approaches for selective encryption (DES, RSA and AES) with AVK and their comparative study" published in Int'l Journal HIT Transactions on ECCN , Vol 1 ,No.4, pp236-244

[11] P.Chakrabarti , B.Bhuyan , A.Chowdhuri , C.T.Bhunia , "Application of Automatic Variable Key (AVK) in RSA" , accepted for publication in Int'l Journal HIT Transactions on ECCN , Vol 2 ,No.5 (in press)

[12] C.T.Bhunia et al , Application of Automatic Variable Key in ECB with DES and RSA , Proc.Annual CSI Conference , Tata McGraw Hill , 2004, pp-135-145

[13] P.Chakrabarti , G.H.Mondal , B.Bhuyan , A.Chowdhuri , C.T.Bhunia "Various New and Modified Approaches for selective encryption with AVK (diffusion and fuzzy) and their comparative study", selected and to be published in IEEE Conference , ITNG08, USA , April'08