# Key Generation in the Light of Mining and Fuzzy Rule

**P. Chakrabarti [1], MIEEE  A. Choudhary [1]  N. Naik [1]  C.T.Bhunia [2], SMIEEE**

[1]Oriental Institute of Science and Technology , Bhopal-462021,Madhya Pradesh,India
[2]Haldia Institute of Technology,Haldia 721657, West Bengal ,India and ICTP, 34014 Trieste , Italy
Main/Corresponding author email_id : **prasun9999@rediffmail.com**

**Summary**

Information security plays a major role in case of secured data transmission . In this paper some intelligent techniques have been pointed out regarding shared key generation. In case of multiparty communication the concept of shared key is essential that therein the security level is increased, as the entire key is not transmitted. Various proposed techniques have been cited based on minimal frequent set, candidate generation, partition scheme, intersection of item-set count. The paper also deals with the efficient generation of shared keys required for direct communication among co-processors without active participation of server. Hence minimization of time-complexity , proper utilization of resource as well as environment for parallel computing can be achieved with higher throughput in secured fashion. The techniques involved are cryptic methods based on feature analysis centroid analysis, inter-centroid distance , extraction scheme of vowel , index position of character , support analysis and confidence rule. We have also proposed several schemes for key evaluation based on theory of central tendencies and curves without transmission of the entire key through the channel thereby making the hacker confused regarding the value of the shared key.

***Key words***

*Shared key, minimum frequent set, candidate generation, partition scheme, item-set count, central tendencies, and curves.*

## 1. Key evaluation based on minimal frequent set

The minimal frequent set can be formed based on the minimum probability of the combination of items. The shared key is the XOR of the XOR values of each of the pairs of elements of the set.

| Message | Keys |
|---|---|
| m1 | SK1 = f($K_1$, $K_3$, $K_4$, $K_6$) |
| m2 | SK2 = f($K_3$, $K_5$) |
| m3 | SK3 = f($K_4$, $K_5$, $K_6$) |
| m4 | SK4 = f($K_2$, $K_3$, $K_5$) |
| m5 | SK5 = f($K_1$, $K_2$) |
| m6 | SK6 = f($K_1$, $K_2$, $K_3$, $K_6$) |

Among the combination of the keys, only ($K_1$, $K_5$) and ($K_2$, $K_4$) have least probability and it is zero.

So, minimal frequent set = {$K_1$, $K_5$, $K_2$, $K_4$}
So, shared key = ($K_1$XOR $K_5$) XOR (K2XOR $K_4$)

## 2. Key evaluation based on candidate generation

| Message | Keys |
|---|---|
| m1 | SK1 = f($K_1$, $K_3$, $K_4$, $K_6$) |
| m2 | SK2 = f($K_3$, $K_5$) |
| m3 | SK3 = f($K_4$, $K_5$, $K_6$) |
| m4 | SK4 = f($K_2$, $K_3$, $K_5$) |
| m5 | SK5 = f($K_1$, $K_2$) |
| m6 | SK6 = f($K_1$, $K_2$, $K_3$, $K_6$) |

Table 1: Key evaluation based on candidate generation

| Candidate No. | Elements | Count | Key Having Minimum-Count |
|---|---|---|---|
| $C_1$ | $K_1$ | 3 | $K_4$ |
| | $K_2$ | 3 | |
| | $K_3$ | 4 | |
| | $K_4$ | 2 | |
| | $K_5$ | 3 | |
| | $K_6$ | 3 | |
| $C_2$ | $K_1, K_2$ | 2 | ($K_1, K_5$) and ($K_2, K_4$) |
| | $K_1, K_3$ | 2 | |
| | $K_1, K_4$ | 1 | |
| | $K_1, K_5$ | 0 | |
| | $K_1, K_6$ | 2 | |
| | $K_2, K_3$ | 2 | |
| | $K_2, K_4$ | 0 | |
| | $K_2, K_5$ | 1 | |
| | $K_2, K_6$ | 1 | |
| | $K_3, K_4$ | 1 | |
| | $K_3, K_5$ | 2 | |
| | $K_3, K_6$ | 2 | |
| | $K_4, K_5$ | 1 | |
| | $K_4, K_6$ | 2 | |
| | $K_5, K_6$ | 1 | |

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

So, shared key = $K_4$XOR ($K_1$XOR$K_5$) XOR ($K_2$XOR$K_4$)

## 3. Key evaluation based on partition scheme

| Message | Keys |
|---|---|
| m1 | SK1 = f($K_1$, $K_3$, $K_4$, $K_6$) |
| m2 | SK2 = f($K_3$, $K_5$) |
| m3 | SK3 = f($K_4$, $K_5$, $K_6$) |
| m4 | SK4 = f($K_2$, $K_3$, $K_5$) |
| m5 | SK5 = f($K_1$, $K_2$) |
| m6 | SK6 = f($K_1$, $K_2$, $K_3$, $K_6$) |

Table 2 : Key evaluation based on partition scheme

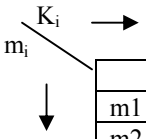
| Candidate No. | Elements | Value | Count | Net value = value * count | Key having Maximum Net value |
|---|---|---|---|---|---|
| $C_1$ | $K_1$ | 0.1 | 3 | 0.3 | $K_6$ |
| | $K_2$ | 0.2 | 3 | 0.6 | |
| | $K_3$ | 0.3 | 4 | 1.2 | |
| | $K_4$ | 0.4 | 2 | 0.8 | |
| | $K_5$ | 0.5 | 3 | 1.5 | |
| | $K_6$ | 0.6 | 3 | 1.8 | |
| $C_2$ | $K_1$, $K_2$ | 0.3 | 2 | 0.6 | ($K_4$,$K_6$) |
| | $K_1$, $K_3$ | 0.4 | 2 | 0.8 | |
| | $K_1$, $K_4$ | 0.5 | 1 | 0.5 | |
| | $K_1$, $K_5$ | 0.6 | 0 | 0 | |
| | $K_1$, $K_6$ | 0.7 | 2 | 1.4 | |
| | $K_2$, $K_3$ | 0.5 | 2 | 1 | |
| | $K_2$, $K_4$ | 0.6 | 0 | 0 | |
| | $K_2$, $K_5$ | 0.7 | 1 | 0.7 | |
| | $K_2$, $K_6$ | 0.8 | 1 | 0.8 | |
| | $K_3$, $K_4$ | 0.7 | 1 | 0.7 | |
| | $K_3$, $K_5$ | 0.8 | 2 | 1.6 | |
| | $K_3$, $K_6$ | 0.9 | 2 | 1.8 | |
| | $K_4$, $K_5$ | 0.9 | 1 | 0.9 | |
| | $K_4$, $K_6$ | 1 | 2 | 2 | |
| | $K_5$, $K_6$ | 1.1 | 1 | 1.1 | |

So, shared key = $K_6$ XOR ($K_4$XOR$K_6$) = $K_4$XOR$K_6$

## 4. Key evaluation based on intersection of item-set count

| Message | Keys |
|---|---|
| m1 | SK1 = f($K_1$, $K_3$, $K_4$, $K_6$) |
| m2 | SK2 = f($K_3$, $K_5$) |
| m3 | SK3 = f($K_4$, $K_5$, $K_6$) |
| m4 | SK4 = f($K_2$, $K_3$, $K_5$) |
| m5 | SK5 = f($K_1$, $K_2$) |
| m6 | SK6 = f($K_1$, $K_2$, $K_3$, $K_6$) |

Table 3 : Key evaluation based on intersection of item-set count




| | $K_1$ | $K_2$ | $K_3$ | $K_4$ | $K_5$ | $K_6$ |
|---|---|---|---|---|---|---|
| m1 | 1 | 0 | 1 | 1 | 0 | 1 |
| m2 | 0 | 0 | 1 | 0 | 1 | 0 |
| m3 | 0 | 0 | 0 | 1 | 1 | 1 |
| m4 | 0 | 1 | 1 | 0 | 1 | 0 |
| m5 | 1 | 1 | 0 | 0 | 0 | 0 |
| m6 | 1 | 1 | 1 | 0 | 0 | 1 |

In first pass, supports counted are:
$\{K_1\}\rightarrow 3$, $\{K_2\}\rightarrow 3$, $\{K_3\}\rightarrow 4$, $\{K_4\}\rightarrow 2$, $\{K_5\}\rightarrow 3$, $\{K_6\}\rightarrow 3$
So, highest = $\{K_3\}$
Ignore $K_6$
In second pass, supports counted are:
$\{K_1, K_2\}\rightarrow 2$, $\{K_1, K_3\}\rightarrow 2$, $\{K_1, K_4\}\rightarrow 1$, $\{K_1, K_5\}\rightarrow 0$, $\{K_2, K_3\}\rightarrow 2$, $\{K_2, K_4\}\rightarrow 0$, $\{K_2, K_5\}\rightarrow 1$, $\{K_1, K_2\}\rightarrow 1$, $\{K_3, K_5\}\rightarrow 2$, $\{K_4, K_5\}\rightarrow 1$
So, highest = $\{(K_1, K_2), (K_2, K_3), (K_3, K_5)\} \equiv \{ K_1, K_2, K_3, K_5 \}$
So, shared key = intersection of two element-sets = $K_3$

## 5. Shared key using feature based method

Let six messages are to be sent by the sender and those have to be encrypted by combination of one or more keys using some function.

| Message | Keys |
|---|---|
| m1 | SK1 = f($K_1$, $K_3$, $K_4$, $K_6$) |
| m2 | SK2 = f($K_3$, $K_5$) |
| m3 | SK3 = f($K_4$, $K_5$, $K_6$) |
| m4 | SK4 = f($K_2$, $K_3$, $K_5$) |
| m5 | SK5 = f($K_1$, $K_2$) |
| m6 | SK6 = f($K_1$, $K_2$, $K_3$, $K_6$) |

Table 4 : Shared key using feature based method

| Key | Initial value | Count | Value | $(Value)^2$ |
|---|---|---|---|---|
| K1 | 0.1 | 3 | 0.3 | 0.09 |
| K2 | 0.2 | 3 | 0.6 | 0.36 |
| K3 | 0.3 | 4 | 1.2 | 1.44 |
| K4 | 0.4 | 2 | 0.8 | 0.64 |
| K5 | 0.5 | 3 | 1.5 | 2.25 |
| K6 | 0.6 | 3 | 1.8 | 3.24 |

Now CF = ( x , y , z )
where x = number of elements , y = linear sum of the elements and z = sum of the square of the elements

CF1 = ( 4 , 4.1 , 5.41)
CF2 = ( 2 , 2.7 , 3.69)
CF3 = ( 3 , 4.1 , 6.13 )
CF4 = ( 3 , 3.3 , 4.05 )
CF5 = ( 2 , 0.9 , 0.45 )
CF6 = ( 4 , 3.9 , 5.13 )

So CFnet = accumulation of maximum of each tuple = ( 4 , 4.1 , 6.13)
So shared key = floor of modulus of (4.1 – 6.13) = 2

## 6. Shared key using centroid based method

CCF = y / x
where x = number of elements , y = linear sum of the elements
CCF1 = 1.025 , CCF2 = 1.35 , CCF3 = 1.37 , CCF4 = 1.1 , CCF5 = 0.45 and CCF6= 0.975
So shared key = floor of ((average of CCF values of each function for each message)*10) = 10

### 6.1 Shared key using inter-centroid distance based method

CCF1 1.025
-0.325
CCF2 1.35 -0.305
-0.02 -0.015
CCF3 1.37 -0.29 -0.105
0.27 0.09 -1.75
CCF4 1.1 -0.38 1.645
0.75 -1.555
CCF5 0.45 1.275
-0.525
CCF6 0.975

So shared key = ceiling of modulus of product of final value and 10 = 18

## 7. Shared key using extraction scheme of vowel :

Table 5 : Value of key for encrypting corresponding message

| Message | Key |
|---|---|
| M1 | K1 = AHE964G |
| M2 | K2 = 14BFX9 |
| M3 | K3 = A8CDM |
| M4 | K4 = 76YKL |
| M5 | K5 = 5BF29QP |

Table 6 : Final result after extraction

| Key | Values extracted | Result after concatenation |
|---|---|---|
| K5 | Q | Q |
| K4 | | Q |
| K3 | A | QA |
| K2 | | QA |
| K1 | A , E | QAAE |

So shared key = QAAE

## 8. Shared key based on index position of character

Table 7 : Corresponding Key Values for encrypting corresponding message

| Message | Key |
|---|---|
| M1 | K1 = AHE964G |
| M2 | K2 = 14BFX9 |
| M3 | K3 = A8CDM |
| M4 | K4 = 76YKL |
| M5 | K5 = 5BF29QP |

Assign values regarding each character of key. In case of alphabet its value will be index position (1-26).

Blank is denoted by 0. In case of double digit for index position , the sum of the digits is taken

Table 8 : Final result after index position analysis

| Key | Values of each character | Number after concatenation of values of each character |
|---|---|---|
| K1 | 1,8,5,9,6,4,7 | 1859647 |
| K2 | 0,1,4,2,6,2+4,9 | 0142669 |
| K3 | 0,0,1,8,3,4,1+3 | 0018344 |
| K4 | 0,0,7,6,2+5,1+1,1+2 | 0076723 |
| K4 | 5,2,6,2,9,1+7,1+6 | 5262987 |

Extract maximum value of each bit and accumulate the result.

Shared key = 5879989

## 9. Communication based on support

### 9.1 Scheme

A and B are two parties . K1,K2,K3,K4,K5,K6 are keys which are protected to A and B only . A sends message m1,m2,m3,m4,m5,m6 in encrypted form with the help of one or more keys . Third party will decipher each message by error-and-trial method and form sets . The key having maximum support is the shared key between A and B . If the number of shared key is more than one then that one is primary while other one is candidate to it .Here we will find shared key so that the third party will not be able to decipher the message.

### 9.2 Mathematical Analysis

| Message | Keys |
|---|---|
| m1 | Sk1= f(k1,k3,k4,k6)=(k1^k3^k4^k6) |
| m2 | Sk2=f(k3,k5)=(k3^k5) |
| m3 | Sk3=f(k4,k5,k6)=(k4^k5^k6) |
| m4 | Sk4=f(k2,k3,k5)=(k2^k3^k5) |
| m5 | Sk5=f(k1,k2)=(k1^k2) |
| m6 | Sk6=f(k1,k2,k3,k6)=(k1^k2^k3^k6) |

So, it is seen that k3 is supported by 4 out of 6 sets of shared key . This support of k3=66.6% . Hence shared key of A& B is k3.
If hacker hacks k1,k2…….,k6 then by applying error-and-trial it will get shared key .
So concept of automatic variable shared key is proposed. The concept is that shared key = (key having maximum support) xor (xor of the value of messages where the support is not available) .

Hence, k3= key having maximum support ,
m3,m5= messages encrypted without k3 .
Therefore , shared key =k3^m3^m5 .
This scheme cannot be revealed to the hacker . So it will hack k3 instead modified value of the shared key.

## 10. Communication based on cent-percent confidence rule

### 10.1 Scheme

Input : m1,m2,m3,m4,m5,m6 to A.
K1,K2,K3,K4,K5,K6 to A and B.

**Step 1**

A encrypts each of the messages with combination of the keys and sends it to B.

**Step 2**

B finds the key which has the confidence level of 100 %,i.e. key1=>key2.

If key1 exists, then key2 will also exist and hence confidence of

Key1=>key2 is 100 %.

**Step 3**

Shared key is key1.

**Step 4**

(Application only for enhancing security level)

Shared (key=key1) XOR (key-new) ,
where key-new can be obtained such that key-new=>key1 is minimum.

### 10.2 Mathematical Analysis

| Message | Keys |
|---|---|
| m1 | Sk1=f(k1,k3,k4,k6)=(k1^k3^k4^k6) |
| m2 | Sk2=f(k3,k5)=(k3^k5) |
| m3 | Sk3=f(k4,k5,k6)=(k4^k5^k6) |
| m4 | Sk4=f(k2,k3,k5)=(k2^k3^k5) |
| m5 | Sk5=f(k1,k2)=(k1^k2) |
| m6 | Sk6=f(k1,k2,k3,k6)=(k1^k2^k3^k6) |

Only k4=>k6 has confidence level of 100 % .
So, shared key=k4(up to step 3).

| Association Scheme | Probability |
|---|---|
| k1=>k4 | 1/3 |
| k2=>k4 | 0 |
| k3=>k4 | 1/4 |
| k5=>k4 | 1/2 |
| k6=>k4 | 2/3 |

So, key-new=k2 since it has least probability . Hence , shared key=k4 XOR k2.

## 11. Key evaluation based on theory of central tendencies

In order to improve the security level, it is proposed that instead of sending the entire key, we can transmit the parameters only. Let, the parameters be x1, x2.

Sender will perform the following steps :
(1) Sense x1,x2
(2) Compute the Geometric mean of the variables. G.M. = $(x1.x2)^{1/2}$
(3) The key shared by the sender will be $(x1.x2)^{1/2}$
Receiver will perform the following steps :
(1) Sense x1,x2

(2) Compute the Arithmetic mean of the variables. A.M. = (x1+x2) / 2
(3) Then find out Harmonic Mean of the variables. H.M. = 2 / ( ( 1/x1 + 1/x2) ) = (2 x1.x2) / (x1+x2)
(4) Finally compute $(A.M.*H.M.)^{1/2}$ and will be equal to $(x1.x2)^{1/2}$
(5) The key shared by the receiver will be $(x1.x2)^{1/2}$

Hence without transmitting entire key , the sender and receiver will communicated with the help of the shared key that they have computed based on two different functions, thereby increasing the security level.

## 12. Key variability based on feature of curves

### 12.1 Based on 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

**Step 1**

$R_1= a + b* d_1 = 1 + 1*d_1$ ,

where a=1=first term of Fibonacci series, b=1=second term, $d_1$=first data. Therefore, cipher 1= $R_1$XOR $k_1$ , where $k_1$= key for first session =$x_1$XOR$y_1$

**Step 2**

$R_2= 2+3*d_2$.

Cipher2= $R_2$XOR$K_3$=$R_2$XOR$x_2$XOR$y_2$

**Step 3**

$R_3=5+8*d_3$.

Cipher3= $R_3$XOR$K_3$= $R_3$XOR$x_3$XOR$y_3$

**Step 4**

$R_4= 13+21*d_4$.

Cipher4= $R_4$XOR$K_4$= $R_4$XOR$x_4$XOR$y_4$

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, cipher1. Therefore, cipher q=$R_1$XOR$K_1$.

Now initially for y=a+ bx , it can get $k_1$, $k_2$, $k_3$, $k_4$So 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 $k_2$=$x_2$XOR$y_2$XOR$d_1$.

### 12.2 Based on parabola

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= $y_1$XOR[$\log_d$e]
where d= $4^{th}$ term is series=3 , e= $5^{th}$ term in series=5
So cipher of previous stage depends on the random number of next stage. If hacker knows random sequence and cipher1, it can get $d_1$So,

Solution is Cipher2= $y_1$XOR[$\log_d$(en)] ,
where n is a number private to sender only.

### 12.3 Based on Gompertz curve

The equation is log y= $a+bc^x$.
R1 = $\log(a+bc^x)$.
Cipher1 = $R_1$XOR[$\log_d$e].

For enhancing security,
cipher1 = $R_1$XOR[$\log_d$en].
Even if hacker knows $R_1$, difficult to find x as inverse logarithmic function takes much time and it is infeasible.

### 12.4 Based on imaginary variable

#### 12.4.1 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-efficient of real and imaginary parts are known to user only.

Advanced security level is applied. $k_2$ of next step= $d_2 XOR k_1$.

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 $(2d_2k_2)$

### 12.4.2 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.

## 13. Conclusion

The paper shows how efficiently shared key can be generated in the light of fuzzy based data mining theory. The proposed techniques and their mathematical analysis also show the viability of artificial intelligence regarding shared key computation in case of message transmission in multi-party domain.

### References

[1] Chakrabarti P. and et.al. (2007), "Shared key evaluation in multiparty communication" published in International Conference on IT, Jabalpur.
[2] Chakrabarti P(2008), "An Intelligent Scheme towards information retrieval" accepted for publication in AJIT (Asian Journal of Information Technology).
[3] Chakrabarti P. and et.al. (2008), "A novel approach towards realizing time variant key in cryptography" published in International Journal of Computer Science and Network Security, Korea , May2008.
[4] Chakrabarti P. and et.al. (2008), "Intelligent scheme of data security and message transmission", published in International conference on Emerging Technologies and Applications in Engineering, Technology and Sciences , Rajkot
[5] Chakrabarti P. and et.al (2008). "Approach towards key generation in multi-party communication and computational complexity of RSA algorithm" published in NCET-08, Integral University, Lucknow.
[6] Pujari A.K. (2001),"Data Mining Techniques", University Press.