# End-End Authentication and Cryptography based on Mobile Number and Mac Address

**S Shunmuga Sundaram**

Sri Jayendra Swamigal Silver Jubilee Matriculation Higher Secondary School, Maharaja Nagar-627 002, Tirunelveli District, Tamil Nadu, India

**Abstract:** ***In this process, End-End Authentication is to be performed. It purely based on Mobile Numbers and MAC Addresses of both Sender and Receiver. It has 3 stages for Authentication, In initial stage, Init-Key is to be created using SMob, RMob, SMacA and RMacA. In stage 2, Key-Mat should be generated randomly as well as Dynamically. In stage 3, Auth-Key is to be created using Key-Mat. For encryption and decryption, to use Auth-Key. It will be very useful for sending messages in secure manner because of End-to-End authentication is to be performed.***

**Keywords:** SMob : Sender's Mobile Number, RMob : Receiver's Mobile Number, SMacA : Sender's MAC Address, RMacA : Receiver's MAC Address, Init-Key : Initial Key, Key-Mat : Key Matrix, Auth-Key : Authentication Key, ENC : Encryption , DEC : Decryption, Enc-File : Encrypted File , Dec-File : Decrypted File

## 1. Introduction

In this Authentication process, involves various stages, Initially the first stage has various steps, The initial step in which to need SMob , SMacA , RMob and RMacA. In second step, separation process will be carried out on the Mobile number of both sender as well as receiver. Similarly separation process will be carried out on MAC address of both. In third step involves merging of each separated Mobile number and MAC address as different manner. In second stage Key-Mat will be generated dynamically based on a digit from RMacA of receiver. In third stage, Auth-Key will be created from dynamically created Key-Mat. In final stage encryption or decryption process will be carried out. it will be purely based on Auth-Key.

## 2. Creation of Authentication Key

Authentication process involves various stages. They are as follows.

### 2.1 Initial Key Creation (Init-Key)

It involves the following steps while on creation of Init-Key as

**Step 1: Choose the Receiver**

In this process to select who is a sender and a receiver. After choosing the both, we can collect Mobile Number and MAC Address of both.

**Step 2: Separation Process**

Separation process involves on Mobile number and MAC address like as follows

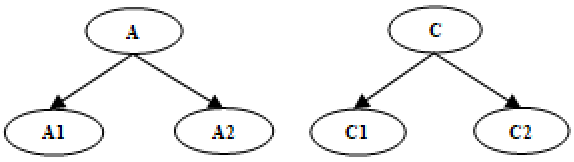

**Figure 1:** Separation of SMob and RMob

Let A and B be the Sender's Mobile Number(SMob) and Sender's MAC Address(SMacA) respectively. Similarly C and D be the Receiver's Mobile Number (RMob) and Receiver's MAC Address (RMacA) respectively.

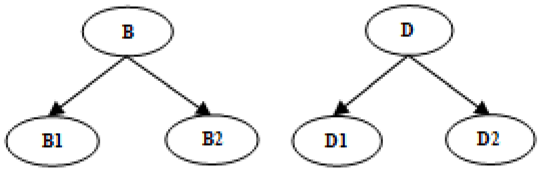

**Figure 2:** Separation of SMacA and RMacA

**Step 3: Merging of SMob and RMob**

With reference to the figures Fig 1 and Fig 2 , merging of SMob and RMob to carried out as and to create intermediate key A3 and A4 as

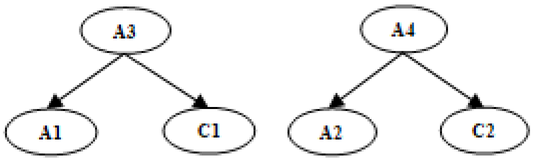

**Figure 3:** Merging of SMob and RMob

**Step 4: Merging of SMacA and RMacA**

With reference to the figures Fig 1 and Fig 2, merging of SMacA and RMacA to be carried out and to create intermediate key B3 and B4 as B3 is obtained as merging of D1 with B1. Similarly B4 is obtaining from merging of D2 with B2.

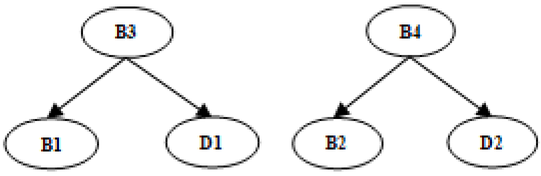

**Figure 4:** Merging of SMacA and RMacA

**Step 5: Obtaining a Partial Key**

The intermediate key A5 and B5 is to be created with the keys A3, A4, B3 and B4 as that is the key A4 is merged with A3 then it will be formed the partial A5, similarly B4 is merged with B3 it will form a partial key known as B5

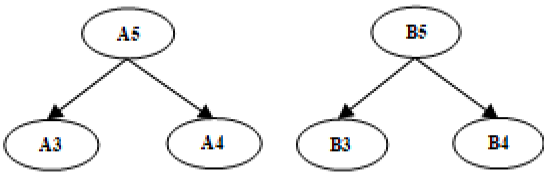

**Figure 5:** Merging of A3 with A4 and B3 with B4

**Step 6: Obtaining an Initial Key [Init-Key]**

The initial key is to be created with the above said partial keys A5 and B5 that is the key A6 is obtained as B5 is merged with A5

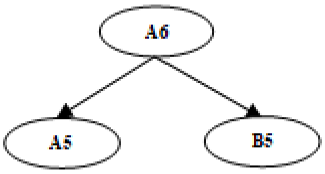

**Figure 6:** Obtaining the Init-Key

### 2.2 Key Matrix Creation (Key-Mat)

**Step 1: Initial Element Selection**

It involves as to choose a digit from RMacA that is it will be considered as starting element of the Key Matrix like as Key-Mat[0][0]. Here considered the digit '7' as starting element then the key martrix as the Fig.7

| | | | | | | |
|---|---|---|---|---|---|---|
| 7 | 8 | 9 | ... | 4 | 5 | 6 |
| 8 | 9 | A | | 5 | 6 | 7 |
| 9 | A | B | ... | 6 | 7 | 8 |
| ⋮ | ⋮ | ⋮ | ... | ⋮ | ⋮ | ⋮ |
| 4 | 5 | 6 | ... | 1 | 2 | 3 |
| 5 | 6 | 7 | ... | 2 | 3 | 4 |
| 6 | 7 | 8 | ... | 3 | 4 | 5 |

**Figure 7:** Key Matrix (16 * 16)

**Step 2: Elements Filling Methods**

After selection of a digit from **RMacA**. Let the digit as **L** then elements of **Key-Mat** calculated as Ref. the above Fig : Sample Key Matrix table as given above.

```
P=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F']
If L Equal to 'A'     Then N=10
Else If L Equal to'B'Then N=11
Else if L Equal to 'C' Then N=12
Else if L Equal to 'D' Then N=13
Else if L Equal to 'E' Then N=14
Else if L Equal to 'F' Then N=15 Else N= L
T=16

For i in Range(0 to 16)
  K=N
  For j in Range(0 to 16)
    Key-Mat[i][j]=P[N]
    If N==T-1 Then N=-1 Else N=N+1
  N=K+1
  if K==T-1 Then N=0
```

### 2.3 Authentication Key Creation (Auth-Key)

While on creation of Auth-Key it involves that Init-Key as well as Key-Mat as such a way

Auth-Key[0]=Init-Key[0][1]
Auth-Key[1]=Init-Key[1][2]
. .
. .
. .
Auth-Key[n]=Init-Key[n][0]

where 'n' is the length of the key 'Init-Key'. For that, the length of Init-Key is same as Auth-Key. Finally, the Auth-Key is useful for encryption of any text file.

## 3. Process of Encryption

In Encryption process, the length of Auth-Key is less than the size of the file 'F' then each byte of Auth-Key is used as round robin method such that

i={0,1,2,…, m} where 'm' is the size of the file
j={0,1,2,…,n} where 'n' is the length of Auth-Key.

and to perform based on condition 1 or 2.

**Condition 1 : Size of the file less than or equal to length of the Auth-Key**

In this stage, the first byte of the file 'F' is encrypted with the first byte of the Auth-Key, second byte of the file encrypted with second byte of Auth-Key and so on upto end byte of the file because the size of the file 'm' is equal with the size of the Auth-Key 'n'. so that

```
For i in range(0..m-1)
    F[i] encrypted with Auth-Key[i]
```

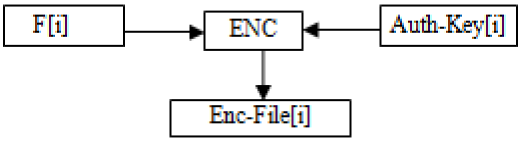

**Figure 8 (a):** Obtaining encrypted file

$$\boxed{\text{Enc-File}[i] = (F[i] + \text{Auth-Key}[i]) \% 256}$$

**Figure 8 (b):** Obtaining encrypted byte

**Condition 2: Size of the file greater than length of the Auth-Key**

In this stage, suppose the size of the file 'm' is greater than the size of Auth-Key then we assume as k=0 , i=0 and perform as

```
Step 1 : To check whether k less than n, if yes
     Step 2 : F[i] is encrypted with Auth-Key[k]
Step 3 : the value of 'k' is incremented that k=k+1
Step 4 : Again to check k equal to n, if yes , to assign k=0,
       i=i+1 else i=i+1
Step 5 : to check 'i' equal to m-1, if yes to perform the step 6
         else to follow the above step 1.
Step 6 : Stop the execution
```

```
Let k=0
   For i in range(0..m-1)
      If k<n then
         F[i] encrypted with Auth-Key[k]
         k=k+1
      If k=n then
         k=0
```

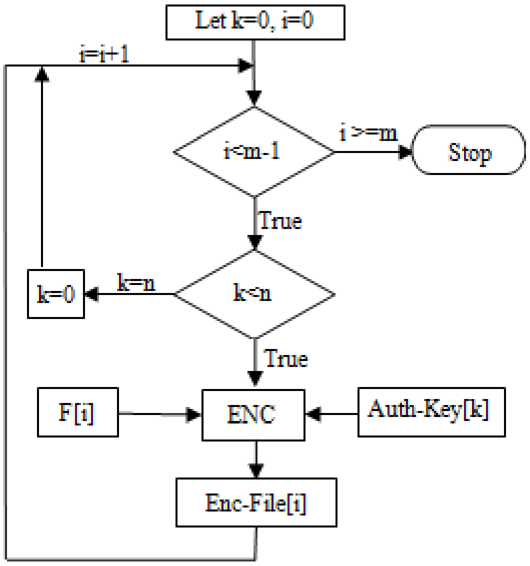

**Figure 8 (c):** Obtaining encrypted file

$$\boxed{\text{Enc-File}[i] = (F[i] + \text{Auth-Key}[k])\ \%\ 256}$$

**Figure 8 (d):** Obtaining encrypted byte

## 4. Process of Decryption

In Decryption process, the length of Auth-Key is less than the size of the file 'F' then each byte of Auth-Key is used as round robin method such that

i={0,1,2,…, m} where 'm' is the size of the file

j={0,1,2,…,n} where 'n' is the length of Auth-Key.

and to perform based on condition 1 or 2

**Condition 1: Size of the file less than or equal to length of the Auth-Key**

In this stage, the first byte of the file 'F' is decrypted with the first byte of the Auth-Key, second byte of the file decrypted with second byte of Auth-Key and so on upto end byte of the file because the size of the file 'm' is equal with the size of the Auth-Key 'n'. so that

```
For i in range(0..m-1)
   F[i] decrypted with Auth-Key[i]
```

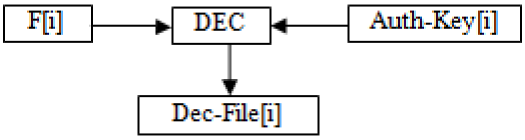

**Figure 9 (a):** Obtaining decrypted file

$$\boxed{\text{Enc-File}[i] = (F[i] - \text{Auth-Key}[i])\ \%\ 256}$$

**Figure 9 (b):** Obtaining decrypted byte

**Condition 2: Size of the file greater than length of the Auth-Key**

In this stage, suppose the size of the file 'm' is greater than the size of Auth-Key then we assume as k=0 , i=0 and perform as

Step 1 : To check whether k less than n, if yes
Step 2 : F[i] is decrypted with Auth-Key[k]
Step 3 : the value of 'k' is incremented that k=k+1
Step 4 : Again to check k equal to n, if yes , to assign k=0,
i=i+1 else i=i+1
Step 5 : to check 'i' equal to m-1, if yes to perform the step 6
else to follow the above step 1.
Step 6 : Stop the execution

```
Let k=0
   For i in range(0..m-1)
      If k<n then
         F[i] decrypted with Auth-Key[k]
         k=k+1
      If k=n then
         k=0
```

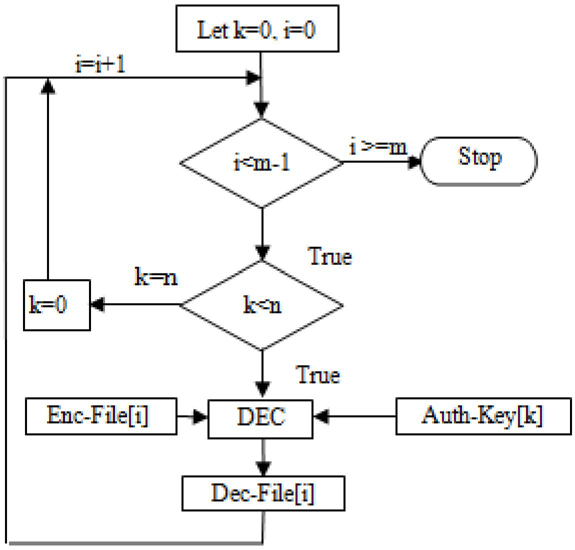

**Figure 9 (c):** Obtaining decrypted file

$$\boxed{\text{Enc-File}[i] = (F[i] - \text{Auth-Key}[k]\ )\ \%\ 256}$$

**Figure 9 (d):** Obtaining decrypted byte

## 5. Working Principles

In need of End to End Authentication to follow as

***Sender's Side:*** To get SMob, RMob and RMacA only and then to perform the following as per order given below

1) *Creation of Authentication Key*

In this process involves MAC address of the sender's system either it may be PC or Laptop in addition with SMob, RMob and RMacA

2) *Process of Encryption*

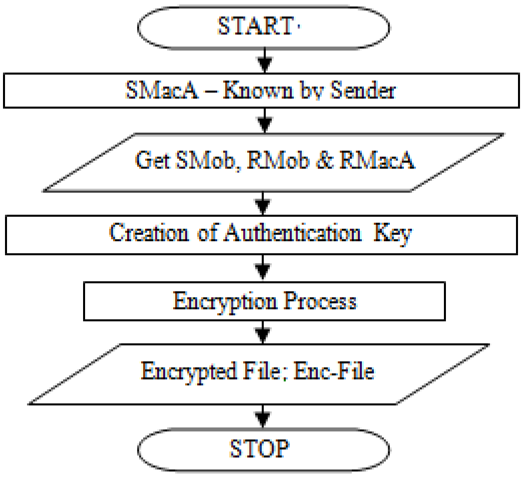

**Figure 10:** Process of Sender

***Receiver's Side:*** To get RMob, SMob and SMacA only then to perform the following as per order given below

1) Creation of Authentication Key
   In this process involves MAC address of the receiver's system either it may be PC or Laptop in addition with RMob, SMob and SMacA
2) Process of Decryption

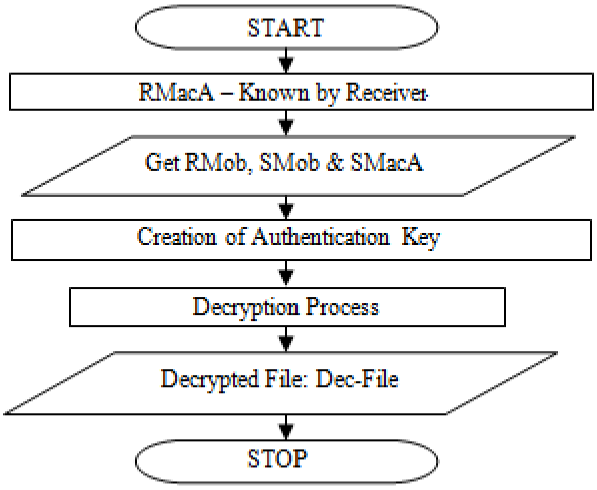

**Figure 11:** Process of Receiver

## 6. Recovery Process

Suppose the system of receiver is in faulty condition either it may in hardware problem or operating system problem, in this case the receiver must follow the procedure for getting the plain text from Enc-File by using third party system.

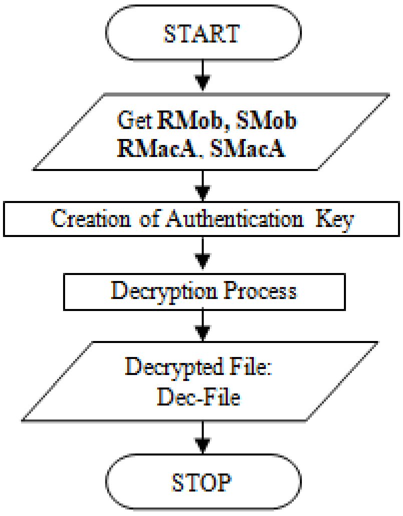

**Figure 12 :** Process of Recovery

## 7. Future Enhancement

Here, considered as one to one authentication only, suppose we want to be sent to more than one authenticated user then it will be modified as slightly and also forwarded to another authenticated user then not known as who is the originator by the third authenticated user that is immediate receiver is considered as sender so we unable to predict who is the source by the third authenticated user.

## Author Profile

**S. Shunmuga Sundaram** received the ***M.E.*** - Computer Science and Engineering from Manonmaniam Sundaranar University, Tirunelveli, April 2011, ***A.M.I.E.*** - Computer Engineering from the Institution of Engineers (India), Kolkata, March 2007 and ***Diploma*** - Computer Technology from Sankar Institute of Polytechnic, Tirunelveli, April 1997and also received Microsoft Certified Professional ***(MCP)*** and IBM Certified Database Associate ***(DB2 Fundamentals)*** and Life Member of the following ***ISTE*** (LM53621),***AMIE*** (AM1327421) and also ***IAEng.*** He stayed in Assistant Professor, Holycross Engineering College Vagaikulam, Tuticorin, Tamil Nadu. System Analyst, National Engineering College, Kovilpatti, Tuticorin, Tamil Nadu. Lab Technician, National Engineering College, Kovilpatti, Tuticorin, Tamil Nadu.. He now working in SJSSJS Matriculation Higher Secondary School, Maharaja Nagar, Tirunelveli-627002, Tamil Nadu.