**Encryption Method’s:-**

In IFIS transaction two types of encryption methods are available to encrypt the finger print.

1. Symmetric Encryption - same key for encryption and decryption.
2. Asymmetric Encryption – different key for encryption and decryption

Encryption Procedure:-

1. Generate the 32 byte random number.(session key)
2. Encrypt the Finger print(ISO Template only) with 32 bit session key using symmetric Encryption Algorithm(AES/ECB/256 PKCS padding 7)
3. Convert the encrypted finger print to BASE64 format and place to DE 63
4. Encrypt the session key using public key extracted from the public certificate which we have to share the vendor’s using Asymmetric Encryption Algorithm(RSA/ECB/ PKCS Padding 1)
5. Convert the encrypted session key to BASE64 format and place to DE 127

**Error U3:-**

Invalid Bio Metric Data

**Error 52:-**

Account Number not in database.

**Error UE:-**

We used to get the UE error for some of issue's

1. Wrong Certificate’s
2. Wrong Asymmetric Encryption
3. If session key size is not in 32 byte’s
4. If Encrypted session key is not in 256 byte’s

How to find the certificate is correct or not?

We can find the encrypted certificate to check the expiry date of certicate

Compare the certificate Expiry Date's with DE 127 Dates what FIGS received from Vendor

Bit: 127 Value=001344TWILG12y2IFk5GCi6vdB/IGzqY0IiXGQOOQhAkxdSZVdfamAtlq8f5qY4Oyw7lfjuHzEdeAh6D39MM/kF9toTXi/14JpD70/znoVdRuKLwXGXzoaWtH8TXaDYwHzmptSOTK458LmTF4aPV6jQToSrLddPyC/Vq2o3sliI73WRgT/mPzmqOOCosolaTBE3oSeNQmexc2QQE8xcf4XYSdRGCeWRr1yRrxA6deb8xhLMnLtK58QJGuYiLu2YQpB+GQKpyNiCG9iARpzgySsXiYRtzeP3R+4PmUMzR9wKluG2H2EFND7r724E4zF6dQU6mGj/xmLOB2OmSS/rT2zymzFNg==**00200820150311**

**# openssl x509 -in CBI-PublicCert.pem -noout -enddate**

notAfter=Mar 11 03:58:30 2015 GMT

How to find the encryption is correct or not?

How to check the session key size is 32 byte or not?

Decrypt the Session Key

Take the encrypted session key from DE 127

#echo"TWILG12y2IFk5GCi6vdB/IGzqY0IiXGQOOQhAkxdSZVdfamAtlq8f5qY4Oyw7lfjuHzEdeAh6D39M/kF9toTXi/14JpD70/znoVdRuKLwXGXzoaWtH8TXaDYwHzmptSOTK458LmTF4aQToSrLddPyC/Vq2o3sliI73WRgT/mPzmqOOCosolaTBE3oSeNQmexc2QQE8xcf4XYSdRGCeWRr1yRrxA6deb8xhLMnLtK58QJGuYiLu2YQpB+GQKpyNiCG9iARpzgySsXiYRtzeP3R+4PmUMzR9wKluG2H2EFND7r724E4zF6dQU6mGj/xmLOB2OmSS/rT2zymzFNg==" | fold -w 64 | openssl enc -base64 -d | openssl rsautl -decrypt -inkey /home/iAS-Server1.0.0.1/CBI-PrivateKey\_pcks8.pem > Check

# ls -ltr | grep Check

-rw-r--r--. 1 root root **32** Dec 13 14:35 Check

How to check the Encrypted session key size is 256 byte’s or not?

Take the encrypted session key from DE 127

#echo"TWILG12y2IFk5GCi6vdB/IGzqY0IiXGQOOQhAkxdSZVdfamAtlq8f5qY4Oyw7lfjuHzEdeAh6D39M/kF9toTXi/14JpD70/znoVdRuKLwXGXzoaWtH8TXaDYwHzmptSOTK458LmTF4aQToSrLddPyC/Vq2o3sliI73WRgT/mPzmqOOCosolaTBE3oSeNQmexc2QQE8xcf4XYSdRGCeWRr1yRrxA6deb8xhLMnLtK58QJGuYiLu2YQpB+GQKpyNiCG9iARpzgySsXiYRtzeP3R+4PmUMzR9wKluG2H2EFND7r724E4zF6dQU6mGj/xmLOB2OmSS/rT2zymzFNg==" | fold -w 64 | openssl enc -base64 –d > EncKey

# ls -ltr EncKey

-rw-r--r--. 1 root root **256** Dec 13 14:50 EncKey

**Error UG:-**

We used to get the UG error for some of issue's

1. Wrong Finger Print Template
2. Wrong symmetric Encryption

How to check the Finger Print Template is correct ot not?

Get the ISO template from vendor and run the script.(Script Attached)

[root@report script]# sh Script/FingerPrintValid\_FMR.sh 00000006949582941\_02.FMR

Status of the Finger Print 00000006949582941\_02.FMR

----------------------------------------------------------

Format identifier :- **FMR**

Version Number :- **20**

Record Length :- **Not-Match**

Device ID :- **0**

Image Size Horizontal :- **256 pixels**

Image Size Vertical :- **320 pixels**

Horizontal Resolution :- **197 pixels-per-cm**

Vertical Resolution :- **197 pixels-per-cm**

Number of Finger :- **1**

Reserved :- **0**

Finger Position :- **0**

View Number and impression type :- **0**

Finger Quality :- **100**

Number of minutiae :- **42**

How to check the wrong symmetric encryption is correct or not?

Confirm to vendor whether they are using symmetric Encryption Algorithm (AES/ECB/256 PKCS padding 7)

Note:-

In default packages of openssl will not support the PKCS7 Padding.

Normal Trouble Shooting:-

How to check the Dataelement of iso message?

Convert the ISO tdr file to Hexdump Value.

[root@uat 20131213-00005]# hexdump -C "991386888446801-15000000334716000009 public-1386888446806-STC.tdr" | awk '$1=" "{print}' | sed 's/|.\*|//g' | tr [a-z] [A-Z] | head

0F 00 00 00 00 17 7C 03 E5 31 32 30 30 F2 38 44

81 08 E0 80 02 00 00 00 00 00 00 00 0E 31 39 35

30 38 35 34 36 30 35 31 34 35 30 38 30 32 37 39

37 35 33 31 30 30 30 30 30 30 30 30 30 30 30 30

30 30 30 30 31 32 31 33 31 30 34 34 32 38 30 30

30 30 30 39 31 36 31 34 32 38 31 32 31 33 36 30

31 32 30 31 39 30 35 30 36 32 30 30 30 32 32 33

33 34 37 31 36 30 30 30 30 30 39 20 20 70 75 62

6C 69 63 43 42 49 30 30 30 30 44 41 31 38 37 30

30 31 43 42 49 20 20 20 20 20 20 20 20 20 20 20

Check the vendor code (go through the document to get the correct bypte)

[root@uat 20131213-00005]# echo $((0x0F))

15

[root@uat 20131213-00005]#

Check the Bank Code(go through the document to get the correct bypte)

[root@uat 20131213-00005]# echo $((0x00000000))

0

[root@uat 20131213-00005]#

Check the Device ID (go through the document to get the correct bypte)

[root@uat 20131213-00005]# echo $((0x177c))

6012

[root@uat 20131213-00005]#

Check the length (go through the document to get the correct bypte)

[root@uat 20131213-00005]# echo $((0x03E5))

997

[root@uat 20131213-00005]#

Check the Message type identifier (go through the document to get the correct bypte)

[root@uat 20131213-00005]# echo "31 32 30 30" | xxd -r -p

1200[root@uat 20131213-00005]#

Check the Primary Bitmap and Secondary Bitmap

[root@uat 20131213-00005]# echo "F2 38 44 81 08 E0 80 02 00 00 00 00 00 00 00 0E" | sed 's/\s//g' | sed 's/^/"16 i 2 o /g' | sed 's/$/ p "/g' | awk '{print"echo "$0"| dc"}' | sh

111100100011100001000100100000010000100011100000100000000000001000000\

00000000000000000000000000000000000000000000000000000001110

We will get the hexdump value in itgs value when its unable to parse, so that time we have to convert the hexdump value to ascii value.

[root@uat TDR]# echo "00 0C 39 31 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30 31 39 31 31 31 33 31 32 35 32 33 31 31 31 31 39 36 30 31 30 32 30 30 30 30 30 30 30 35 35 33 38 35 34 39 32 35 30 30 30 34 30 35 30 30 30 30 30 30 30 32 35 30 30 30 34 30 35 33 35 36 31 37 30 30 30 30 30 30 30 32 39 35 38 32 35 37 35 31 32 30 32 33 32 35 7C 35 30 38 35 34 36 7C 36 30 31 30 7C 32 35 30 30 30 34 30 35 30 30 38 31 39 31 31 32 30 31 33" | xxd -r -p

9100000000000000001911131252311119601020000000553854925000405000000025000405356170000000295825751202325|508546|6010|2500040500819112013

[root@uat TDR]#