3GPP TSG SA WG3 Security — SA3#33 **S3-040249**
10-14 May 2004
Beijing, China

Source: **Axalto**

Title: **MBMS key Management comparison**

Document for: **Discussion and decision**

Agenda Item: **MBMS**

1 Introduction

Two key distribution architectures have been proposed for MBMS key management (MSK management).

The first one uses Remote UICC management to update the MBMS keys & files ([1], [2]). The second one is based in GBA and MIKEY protocols run between the BMSC and the ME ([4], [5]).

This contribution analyses both schemas based in network/radio resource consumption.

2 Overview & Motivation

MBMS is introduced on 3GPP with one essential motivation: To enable new data services that make a more efficient usage of the radio spectrum, decreasing the amount of data within the network and using radio resources more efficiently.

It seems then reasonable to compare the two key distribution methods based in their contribution to this basic MBMS requirement.

Four main reasons make the MSK key distribution quite critical in terms of resource usage:

1- It is performed in point to point basis

2- Likely, the number of users will be big

3- the number of subscribed MBMS services per user can also be significant

4- the renewal of keys will be probably performed quite frequently (likely, from one to several months)

These 4 multiplicative reasons make rational to reduce the data sent over the radio interface as much as possible. In this context, estimations of the needed data flows in MSK key delivery could be quite useful.

As a simple approach to this comparison, let us analyse in terms of size, the application level data sent in one MSK update procedure for the two MBMS key management proposals (Without taking into account transport/session headers and signalling and other extra overheads).

3 Comparison

3.1 Messages

The following data flows are taken into account for each of the two proposals:

GBA-MIKEY

A) Initial HTTP request from UE to BMSC over Ua Interface

Containing an HTTP request and rejection response to/from the BMSC.

B) GBA messages over Ub interface (between UE and BSF)

Containing an HTTP Digest AKA dialog between the UE and the BSF. This is used to establish a MUK between BMSC and either the UICC or the ME. This part may be skipped if a new Ks or MUK is not needed.

C) Protected HTTP request messages over Ua Interface

Containing protected MIKEY payload

A possible example of these messages is given in ANNEXE 1 & 2

OTA approach

D) Secure packets containing APDUs (transported either over SMS or GPRS&CATP)

An example of format of this message is given in ANNEXE 3

3.2 Size estimation

The proposal of this section is to have an average size of the data exchanged between the UE and the Network for MSK management in both proposals.

For A, B and C messages (except MIKEY payload) in GBA-MIKEY proposal an approximate value is computed using examples taken from draft TS 24.109 and copied in the ANNEXE 1 of this contribution. To be noted that examples are referred to a NAF acting as PKI portal, but they are likely similar to those when NAF is the BMSC.

An estimation of MIKEY payload size is considered. (The accuracy of this estimation could be confirmed by the companies supporting GBA-MIKEY solution since modifications of basic MIKEY payload, including new MBMS specific extension payloads are being proposed).

GBA-MIKEY

Message	SIZE (in bytes) estimation	Comments
A) INITIAL HTTP REQUEST	170	
A) INITIAL HTTP RESPONSE (401 Unauthorized response)	270	
B1) INITIAL GET	190	
B2) 401 Unauthorized response	240	
B3) HTTP GET request (with the Digest AKA RES)	480	
B4) 200 OK response	250	
C1) GET request	450	
C2) GET RESPONSE (without MIKEY Payload)	280	
C2') MIKEY PAYLOAD	60	Containing HEADER/ Extension / KEMAC &

		Time Stamps payloads
TOTAL (A+C)	**1230**	Not including MUK/ Ks_Naf delivery
TOTAL (A+B+C)	**2390**	

OTA approach (D)

Secured Packet	**SIZE (in bytes) estimation**	**Comments**
1st Security Header	24	
Remote APDU commands (size=Header(5) + Data)		
SELECT	5 + 4	Selection by path : DF $_{MBMS}$ + EF $_{MBMSDescription}$
UPDATE RECORD	5 + 7	Update of MSK_ID, MSK Reference, MSK_Exp, MSK_SEQ
2nd Security Header	24	
Remote APDU commands (size=Header(5) + Data)		
PUT KEY	5 +16 + 8	MSK + key MAC
TOTAL	**98**	

Extra considerations:

- SEVERAL UPDATES in the same packet:

  From several contributions, it seems possible to carry out several MSKs updates in the same MIKEY packet. In the same way, it is completely possible to carry out multiple key/file updates in the same secured packet.

  This could reduce some overhead in header fields.

  This procedure seems anyway more feasible in the case where the updates are managed by the network (OTA) than in the cases that management requests are mobile-originated (probably based in joining procedure or in some key expiration policy) as in GBA-MIKEY approach.

- Synchronization failures:

  They are not taken into account in GBA exchange example. They may increase the data in section B.

- More MIKEY payloads

  More MIKEY payloads are probably needed (e.g. MBMS_ID, MTK_SEQ). Additionally, some interesting features are not yet supported by MIKEY and its corresponding extensions e.g. MSK Deletion. They may likely involve increase in MIKEY message.

4 Conclusion

Comparing the two MSK management proposals in terms of key management traffic, it is considered that OTA based approach is much more efficient (ratio ~ 2000 / 100) given the same or even more functionalities than the GBA/MIKEY solution.

GBA/MIKEY may contradict the main assumption of MBMS when wasting (20 times more) valuable radio resources for MSK key management.

It is proposed to choose the OTA key delivery solution for MBMS.

5 References

[1] S3-040050: MBMS UICC-based solution (Gemplus, Axalto, Giesecke & Devrient and Oberthur)

[2] S3-040051/88: Discussion paper on MBMS key management (Axalto, Gemplus, Axalto, Giesecke & Devrient and Oberthur)

[3] MIKEY: Multimedia Internet KEYing, draft-ietf-msec-mikey-08.txt

[4] TD S3-040059, Enhanced MIKEY in MBMS, SA2#32, Ericsson

[5] TD S3-040081, MIKEY in MBMS, SA2#32, Nokia

[6] ETSI TS 102 225, “Secured Packet structure for UICC based applications”, Rel-6

[7] ETSI TS 102 226, “Remote APDU Structure for UICC based applications”, Rel-6

ANNEXE 1: GBA Examples (from TS 24.109v001)

UE TO NAF

A)

```
GET / HTTP/1.1
Host: naf1.home1.net:1234
User-Agent: NAF1 Applicatino Agent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referrer: http://naf1.home1.net:1234/service
```

```
HTTP/1.1 401 Unauthorized
Server: Apache/1.3.22 (Unix) mod_perl/1.27
Date: Thu, 08 Jan 2004 10:50:35 GMT
WWW-Authenticate: Digest realm="3GPP-bootstrapping@naf.home1.net",
nonce="6629fae49393a05397450978507c4ef1", algorithm=MD5, qop="auth,auth-int",
opaque="5ccc069c403ebaf9f0171e9517f30e41"
```

UE TO BSF

B)

```
GET / HTTP/1.1
Host: registrar.home1.net:9999
User-Agent: Bootstrapping Client Agent; Release-6
Date: Thu, 08 Jan 2004 10:13:17 GMT
Accept: */*
Referer: http://pki-portal.home1.net:2311/pkip/enroll
```

```
HTTP/1.1 401 Unauthorized
Server: Bootstrapping Server; Release-6
Date: Thu, 08 Jan 2004 10:13:17 GMT
WWW-Authenticate: Digest realm="registrar.home1.net", nonce= base64(RAND + AUTN + server specific
data), algorithm=AKAv1-MD5, qop="auth-int"
```

```
GET / HTTP/1.1
Host: registrar.home1.net:9999
User-Agent: Bootstrapping Client Agent; Release-6
Date: Thu, 08 Jan 2004 10:13:18 GMT
Accept: */*
Referer: http://pki-portal.home1.net:2311/pkip/enroll
Authorization: Digest username="user1_private@home1.net", realm="registrar.home1.net",
nonce=base64(RAND + AUTN + server specific data), uri="/", qop=auth-int, nc=00000001,
cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1,
opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=AKAv1-MD5
```

```
HTTP/1.1 200 OK
Server: Bootstrapping Server; Release-6
Content-Type:
Content-Length:
Authentication-Info: qop=auth-int, rspauth="6629fae49394a05397450978507c4ef1",
cnonce="6629fae49393a05397450978507c4ef1", nc=00000001
Date: Expires: Thu, 08 Jan 2004 10:23:17 GMT
```

UE TO NAF

C)

```
GET / HTTP/1.1
Host: naf1.home1.net:1234
User-Agent: NAF1 Applicatino Agent; Release-6
Date: Thu, 08 Jan 2004 10:50:35 GMT
Accept: */*
Referer: http://naf1.home1.net:1234/service
Authorization: Digest username="base64(TID)", realm="3GPP-bootstrapping@naf.home1.net",
nonce="a6332ffd2d234==", uri="/", qop=auth-int, nc=00000001,
cnonce="6629fae49393a05397450978507c4ef1", response="6629fae49393a05397450978507c4ef1,
opaque="5ccc069c403ebaf9f0171e9517f30e41", algorithm=MD5
```

```
HTTP/1.1 200 OK
Server: Apache/1.3.22 (Unix) mod_perl/1.27Content-Type: text/html
Content-Length: 1234
Authentication-Info: qop=auth-int, rspauth="6629fae49394a05397450978507c4ef1",
cnonce="6629fae49393a05397450978507c4ef1", nc=00000001
Date: Thu, 08 Jan 2004 10:50:35 GMT
Expires: Fri, 09 Jan 2004 10:50:36 GMT
<SERVER PAYLOAD>
```

ANNEXE 2: MIKEY PAYLOADS (from [3] and [4])

Header Payload

HEADER PAYLOAD

Extension Payload

Next Payload	**Type**	**Length**
MUK ID		**MSK ID**
MSK Fetch Point		

KEMAC Payload

Next Payload	**ENCR ALGO**	**Length**
Encr data:MSK		
MAC Algo	**MAC**	

Time Stamp Payload

Next Payload	**TS type**	**TS value**

ANNEXE 3: SECURE PACKET CONTENT ([6])

A/Security Header (Command Header)

CPI	CPL	CHI	CHL	SPI	KIc	KID	TAR	CNTR	PCNTR	RC/ CC/DS

B/Secure Data:

Remote command APDU	Remote command APDU	...	Remote command APDU

Remote command coding:

Class byte (CLA)	Instruction code (INS)	P1	P2	P3	Data