3GPP TSG SA WG3 Security — S3#32 **S3-040059**
09 - 13 February 2004
Edinburgh, Scotland, UK

Agenda Item: MBMS
Source: Ericsson

Title: Enhanced MIKEY in MBMS key management

Document for: Discussion/Decision

1. Introduction

In SA3 Ad hoc meeting in Antwerp a concern was raised against SRTP (Secure RTP) [1] that it may suffer from session key pre-calculation problem. Ericsson studied the issue for SA3#31 and presented a solution in [2] that uses MIKEY and mitigates the problem.

In SA3#31 meeting SA3 decided upon the following:

"It will be possible to run the whole MBMS security with ME only, but will also be possible to run key management using the UICC. A migratory path between the two solutions is needed and the solutions will be developed to allow this. Deviations between the two solutions would only be made for the benefit of the whole system (this implies the use of a 2-tiered system). The difference between the two solutions for delivering the low-level keys would be visible only inside the UE and secondly, the BMSC would know which solution is implemented in the UE side. A Rel-6 compliant UE will support both UICC based and ME based solutions and the Operator will have control over the choice of method used for MBMS services.

For the ME part, GBA and MIKEY (with possible 3GPP-specific enhancements, e.g. for the support of encrypted keys) will be used as a basis for the standardised solution. This does not rule out DRM based solutions, e.g. DOWNLOAD."

In this contribution Ericsson presents a key management mechanism that fulfils the agreement achieved in SA3. The mechanism is a further enhancement of the earlier presented paper [2] submitted to SA3#31. It should be noted that the presented mechanism is compatible with MIKEY [3] in the sense that the approved MIKEY internet draft (which has reached RFC status) allows extensions, which however require a new internet draft. The proposed changes to TS 33.2246 are presented in companion pseudo CRs [5] and [6].

2. Two-tiered MIKEY mechanism

The current MIKEY allows extensions to be specified which enable MIKEY to support two-tiered keying mechanism. . The needed extension to MIKEY is basically that

- A new input is defined in the key derivation function. This is the new random value
- The new random value is transported in security protocol to the receiver

The principle of two-tiered MIKEY mechanism is depicted in figure 1. The key management calculates a fresh TEK from the BAK and RAND always when requested by the security protocol. The security protocol (for example SRTP) detects the need for a new TEK when the concatenated value of BAK-id and RAND has changed in the data packet. If the value is unchanged, security protocol does not need an updated TEK.

In case of UICC based solution, the key management function does not need to be in UICC as a whole. It is sufficient that the BAK storage and TEK derivation are in UICC. However, as is required by requirement 5h in [4], the UICC needs to validate the freshness and authenticity of the RAND.

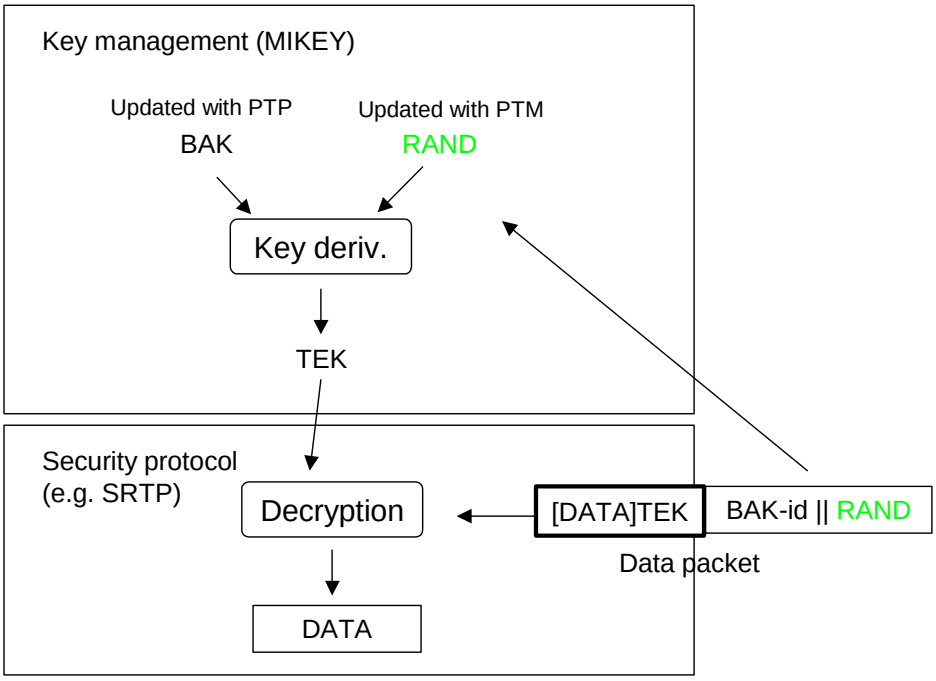

Figure 1 Two-tiered MIKEY mechanism

The mechanism from the UE perspective is depicted in figure 2. It shows also the migration from ME based solution to UICC based solution. UICC based scenario is shown on the left side and ME based scenario is shown on the right side. The figure shows MIKEY as key management protocol and SRTP as an example of a security protocol.

The ME based solution works as follows:

- The KEK may be generated from key material received from GBA.
- The BAK is received in ME by point-to-point key delivery. The BAK is protected with KEK. The lifetime of BAK and BAK-ID are outside the KEK protected content. This is because these parameters are needed by the ME also in the UICC based solution for example to trigger re-keying. Current MIKEY does not support these parameters, but they can be carried in MIKEY in general extension payloads, which can be defined without IETF standardisation.
- User data is received in the ME in point-to-multipoint. If the master key identifier (MKI), i.e. concatenation of BAK-ID and RAND, is new to the security protocol, it requests for a new TEK from the key management protocol. Note that the security protocol does not need to interpret the structure of MKI.
- The key management protocol calculates the new TEK from BAK and RAND and provides it to the security protocol.
- The security protocol decrypts the data
- If the MKI is unchanged, the SRTP can decrypt the data directly.

The UICC based solution is as follows:

- The provision of KEK to the UICC is FFS.
- The MIKEY message is received by point-to-point key delivery in ME. The ME sends the encrypted part of MIKEY message to the UICC, which decrypts the BAK. The lifetime of BAK and BAK-ID are outside the KEK protected content. This is because these parameters are needed by the ME also in the UICC based solution for example to trigger re-keying. Current MIKEY does not support these parameters, but they can be carried in MIKEY in general extension payloads, which can be defined without IETF standardisation.

- User data is received in the ME in point-to-multipoint. If the master key identifier (MKI), i.e. concatenation of BAK-ID and RAND, is new to the security protocol, it requests for a new TEK from the key management protocol.
- The key management protocol checks the BAK-ID and requests for a new TEK from the UICC. If the BAK-ID was not found in ME, it may trigger re-keying to request for the current BAK from the BM-SC.
- The UICC calculates the new TEK from BAK and RAND and provides it to the key management protocol, which forwards the TEK to the security protocol.
- The security protocol decrypts the data
- If the MKI is unchanged, the SRTP can decrypt the data directly.

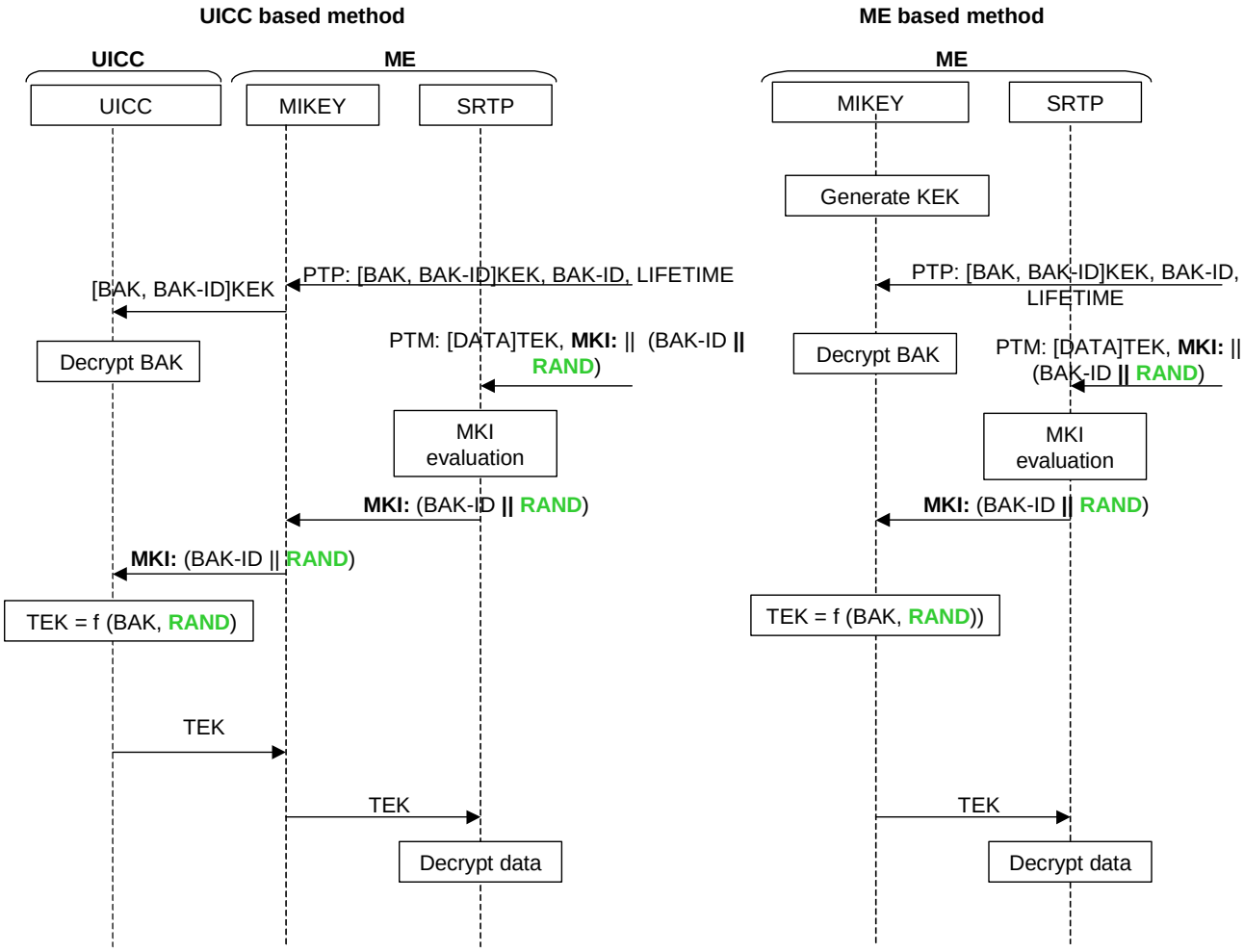

Figure 2 The UICC based solution and ME based solution

3. Load balancing in key requests

Key lifetime

Several earlier contributions have noted the problem that UEs should not request the new MBMS key simultaneously, since this might lead to overload situation. This is also captured in an editor's note in TS 33.246 in 5.2 [4]:

Editor's note: If all users need to request a key update simultaneously then there may need to be some method of ensuring that all the users do not request a key update at the same time. This mechanism is ffs.

An important parameter in avoiding this problem is the key lifetime. The UEs can for example spread the key requests randomly over the lifetime of the key. If the UEs have no knowledge of the key lifetime, they will most probably request the key when it changes in the multicast data, thereby causing a peak of key requests.

There is currently no requirement of the key lifetime parameter associated to high level MBMS key despite its evident importance. Therefore it is proposed to add the following requirement to TS 33.246:

There shall be a lifetime value associated to a high MBMS level key in order to support mechanisms that prevent all UEs from requesting the new high- level key simultaneously. This key lifetime shall be communicated to the UE with the associated key.

The exact change is included in companion pseudo CR [5].

Batch of keys

It is seen beneficial to be able to send many keys to the UE in one batch. This can for example decrease the key signalling load, decrease the risk for peaks in key requests and also enable various charging schemes. Therefore it is proposed to add the following requirement to TS 33.246:

It shall be possible for the keying mechanism to send many keys (a batch of keys) with one keying message to the UE in order to decrease signalling load in the network and to support load balancing in key requests and different possible charging schemes.

The exact change is included in companion pseudo CR [5].

MIKEY support

MIKEY can support both of the above requirements. Already the current MIKEY can carry several encrypted BAKs in one MIKEY message. The key-id *or* the lifetime of the key is present in the encrypted key payload as an optional element.

The lifetime associated with a key-id can also be added to MIKEY outside the encrypted part in general extension payload by defining a new TLV, e.g. as [key-id, lifetime]. A MIKEY message may include as many of these payloads as there are keys in the message, if a batch of keys is sent. The general extension payload is also protected by MIKEY MAC.

A regular MIKEY message with one key inside the encrypted KEMAC part is as follows, see [2] chapter 3.1:

MIKEY message = HDR, T, RAND, [IDi], {SP}, KEMAC (Key-id-1, Key-1),

where KEMAC (Key-1, Key-id-1) means that Key-1 with its Key-id-1 are carried in the encrypted KEMAC part.

When several keys are sent in one message and general extension payloads are used to convey the key-ids and lifetimes outside the encrypted part, the message could look as follows:

MIKEY message = HDR, T, RAND, [IDi], {SP}, **GEP (Key-id-1, Life-1), GEP (Key-id-2, Life-2)**, KEMAC (Key-id-1, Key-1, Key-id-2, Key-2),

where GEP is general extension payload, which can carry the key-id and lifetime for a key. GEPs are added before the KEMAC since it includes the MAC for the whole MIKEY message. Note that the Key-id-x inside the KEMAC is optional in MIKEY.

4. IETF considerations

MIKEY internet draft has been approved to be an IETF RFC although an RFC number has not been allocated yet. The approved MIKEY allows certain extensions to be made to MIKEY with new standards track RFCs. The MIKEY enhancement described above to support two-tiered keying mechanism is such an extension. However, the timeframe of release 6 might not allow for new standards track RFC to be written. Therefore Ericsson proposes that the needed extensions to MIKEY should be specified in 3GPP.

5. Migration

The proposed two-tiered MIKEY provides smooth migration path to UICC based method, since it can be used with both ME and UICC based methods.

6. Proposal

It is proposed to adopt two-tiered MIKEY as key management protocol for MBMS. MIKEY can support both ME and UICC based methods. MIKEY also provides smooth migration path to UICC based method. It is also proposed that SA3 endorses the requirements in chapter 3.

7. Conclusion

MIKEY enables both ME and UICC based key management. This document and accompanying pseudo-CR [6] describes how MIKEY is used in ME and UICC based key management.

MIKEY internet draft has been approved to be an IETF RFC although an RFC number has not been allocated yet. The approved MIKEY allows certain extensions to be made to MIKEY with new standards track RFCs. The MIKEY enhancement described above to support two-tiered keying mechanism is such an extension. However, the timeframe of release 6 might not allow for new standards track RFC to be written. Therefore Ericsson proposes that the needed extensions to MIKEY should be specified in 3GPP.

8. References

[1] The Secure Real-time Transport Protocol, draft-ietf-avt-srtp-09.txt, work in progress

[2] TD S3-030723 Migration of MIKEY in MBMS key management, SA3#31 November, Ericsson

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

[4] TS 33.246, Security of Multimedia Broadcast/Multicast Service, v 1.0.0

[5] TD S3-040xxx, Support of key lifetime and batch of keys, SA3#32, February 2004, Ericsson

[6] TD S3-040xxx, ME based and UICC based MBMS key management with MIKEY, SA3#32, February 2004, Ericsson

3GPP TSG-SA3 Meeting #32
Edinburgh, Scotland, 9 - 13 February 2004

Tdoc ⌘S3-04xxxx

CR-Form-v7

CHANGE REQUEST

⌘ **TS 33.246 CR CRNum** ⌘**rev** **-** ⌘ Current version: 1.0.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [X] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	ME and UICC based MBMS Key management with MIKEY
Source:	⌘	Ericsson
Work item code:	⌘	MBMS
Date:	⌘	30/01/2004
Category:	⌘	**C**
Release:	⌘	Rel 6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Key management has not been specified
Summary of change:	⌘	ME and UICC based MBMS key management is performed with MIKEY protocol
Consequences if not approved:	⌘	

Clauses affected:	⌘	6

	Y	N		
Other specs Affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6 Security mechanisms

6.1 Authentication and authorisation of a user

Editor's note: This section will contain the details of how a user joins a particular Multicast Service.

6.2 Key update procedure

Once a UE has joined a multicast service, the UE should try to get the high level key that will be used to 'protect' the data transmitted as part of this multicast service. If the UE fails to get hold of this key or receives confirmation that no updated key is necessary or available at this time, then, unless the UE has a still-valid, older key, the UE shall leave the MBMS user service. The UE tries to get the high level key using the second message in the below flow.

The BM-SC controls when the high level keys used in a multicast service are to be changed. The flow in figure 2 describes how the high-level key changes are performed.

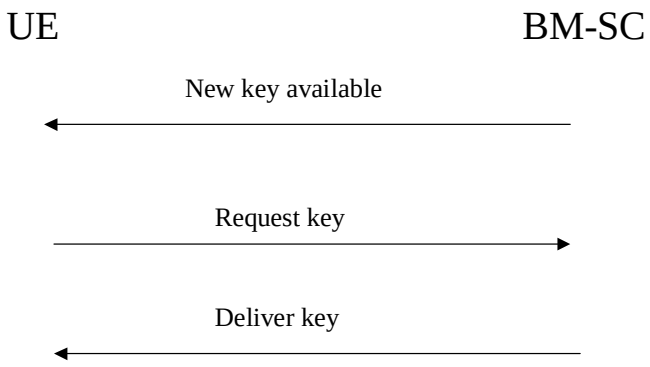

Figure 2: high-level key changes

The first message is sent out by the BM-SC to indicate that new keys are available. It is an optional message in the flow. If it is sent to all UEs, then it needs to be ensured that all the UEs do not request the new key simultaneously.

The second message is used to request a key. This is sent by the UE when it either receives the first message in the flow and does not have the new key, has just joined a multicasts service and does not have a key for that service or a UE has received some protected content which it does key that was used to protect the content. If the UE fails to get hold of the updated key or receive confirmation that no updated key is necessary or available at this time, then, unless the UE has a still valid older key, the UE shall leave the MBMS service.

After receiving the second message the BM-SC should send out the appropriate key to the UE protected by the relevant means. Upon successfully receiving the new key, the UE should store this key for later use.

Editor's note: MIKEY is being considered as the method for carrying keys. Possible optimisations were proposed at the ad-hoc in Antwerp (S3z030010). One identified issue was the possible need to terminate MIKEY in the UICC and/or terminal in the combined method. The use of MIKEY relates to the PTP delivery of a key.

~~6.2~~6.3 ME based ~~K~~key management

6.3.1 Overview

The ME based MBMS key management architecture in depicted in figure 2.

- The MA is a MBMS application in BM-SC that uses key management services of MIKEY protocol.

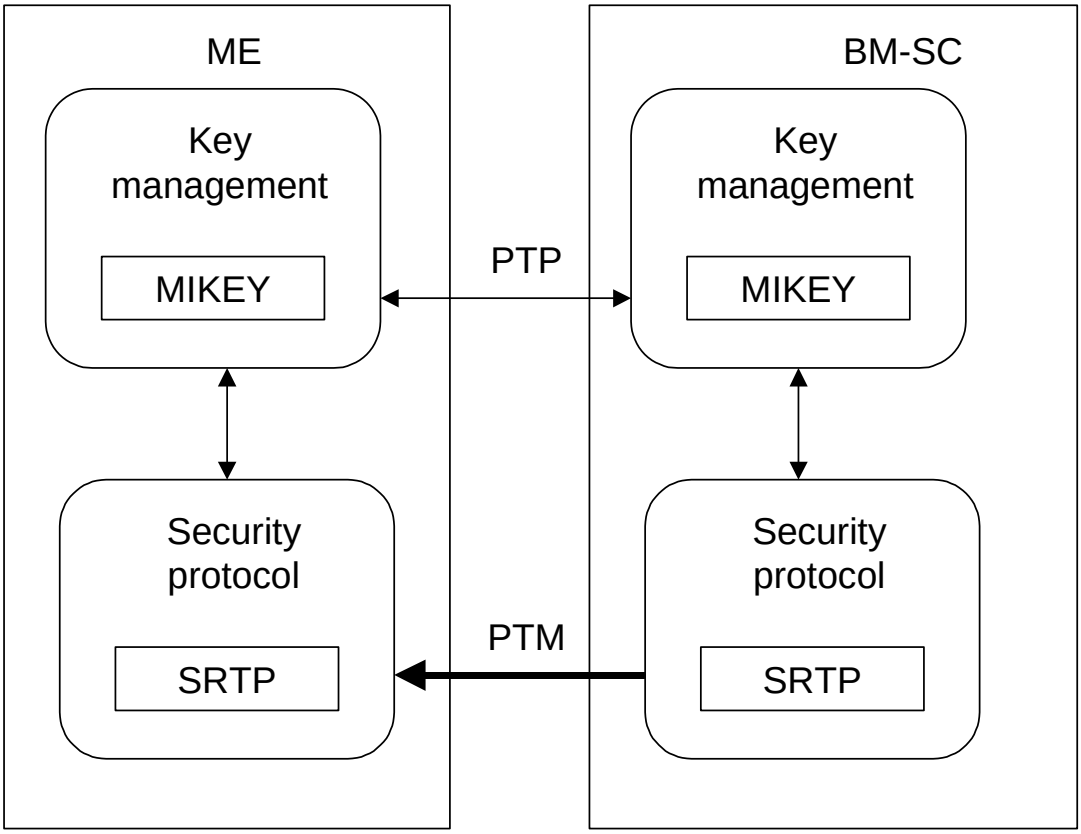

Figure 2: ME based MBMS key management

6.3.2 BAK generation

The MA generates a value for Broadcast Access Key (BAK) and a value for RAND. These actions are independent of the actions of individual UEs.

6.3.3 Point to point initial keying procedure

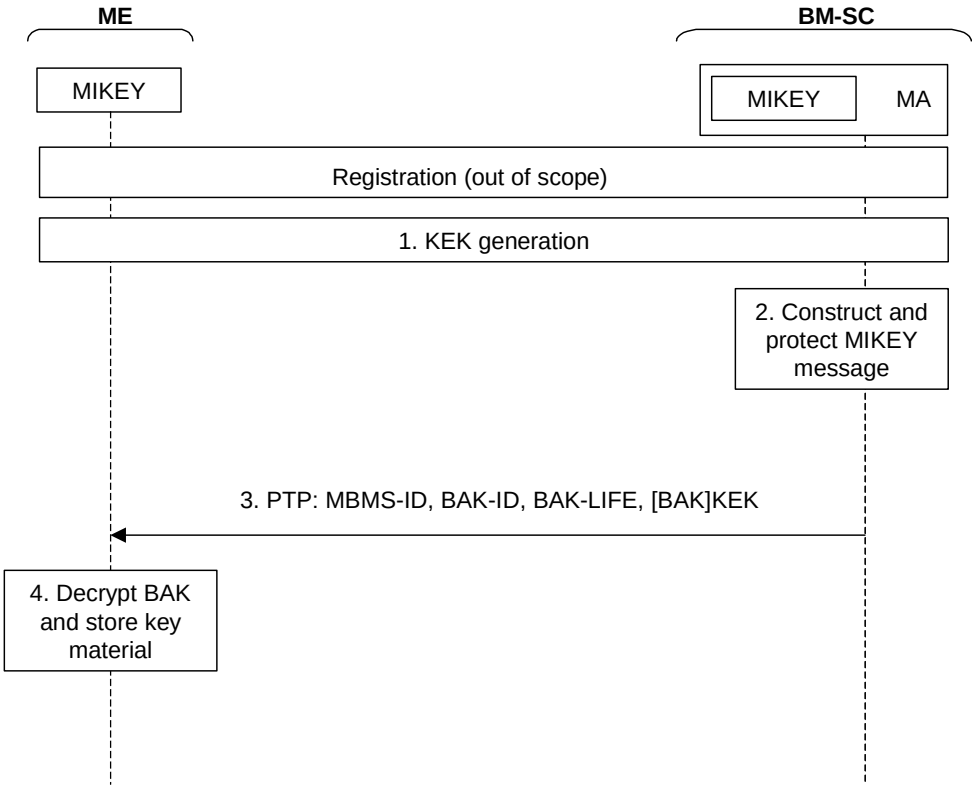

Figure 3: Initial keying procedure. Note that not all parameters are shown in the figure.

Before the initial key exchange is performed, the user has been registered to the service and he/she has been authenticated and authorised for the requested MBMS service.

Editor's note: The authentication of the user is FFS and it could be done e.g. with GBA. Therefore it is out of the scope of current key management description.

1. MA in BM-SC and the ME are provisioned with Key encryption key (KEK). The KEK is used to protect the MIKEY message delivery.

   Editor's note: It is FFS how the KEK is generated, but it may be derived e.g. from the GBA keys. Also MIKEY can be used to generate the KEK from pre-shared keying material.

2. MIKEY message is constructed in BM-SC: the MBMS service ID that identifies the service is included into CSB field in MIKEY header, the BAK is encrypted with KEK, the BAK lifetime (BAK-LIFE) and BAK identifier (BAK-ID) are included as general extension payload field in MIKEY message.
   Note that MIKEY has MAC checksum that covers the whole MIKEY message and a timestamp to protect against replay attacks.

   The MIKEY message is as follows [MIKEY]:

   MIKEY message = HDR, T, RAND, [IDi], {SP}, GEP (Key-id-1, Life-1), KEMAC (Key-1),

   where GEP is general extension payload, which can carry the key-id and lifetime for a key. GEP are added before the KEMAC since KEMAC includes the MAC for the whole MIKEY message. The Key-id-x inside the KEMAC is optional.

3. BM-SC sends the MIKEY message to the UE is point-to-point manner.

4. UE receives the MIKEY message and verifies the timestamp and MAC checksum. If they are correct, the message is processed. UE decrypts the BAK with KEK and stores the key material.

6.2.3 Point to Multipoint re-keying and data transmission procedure

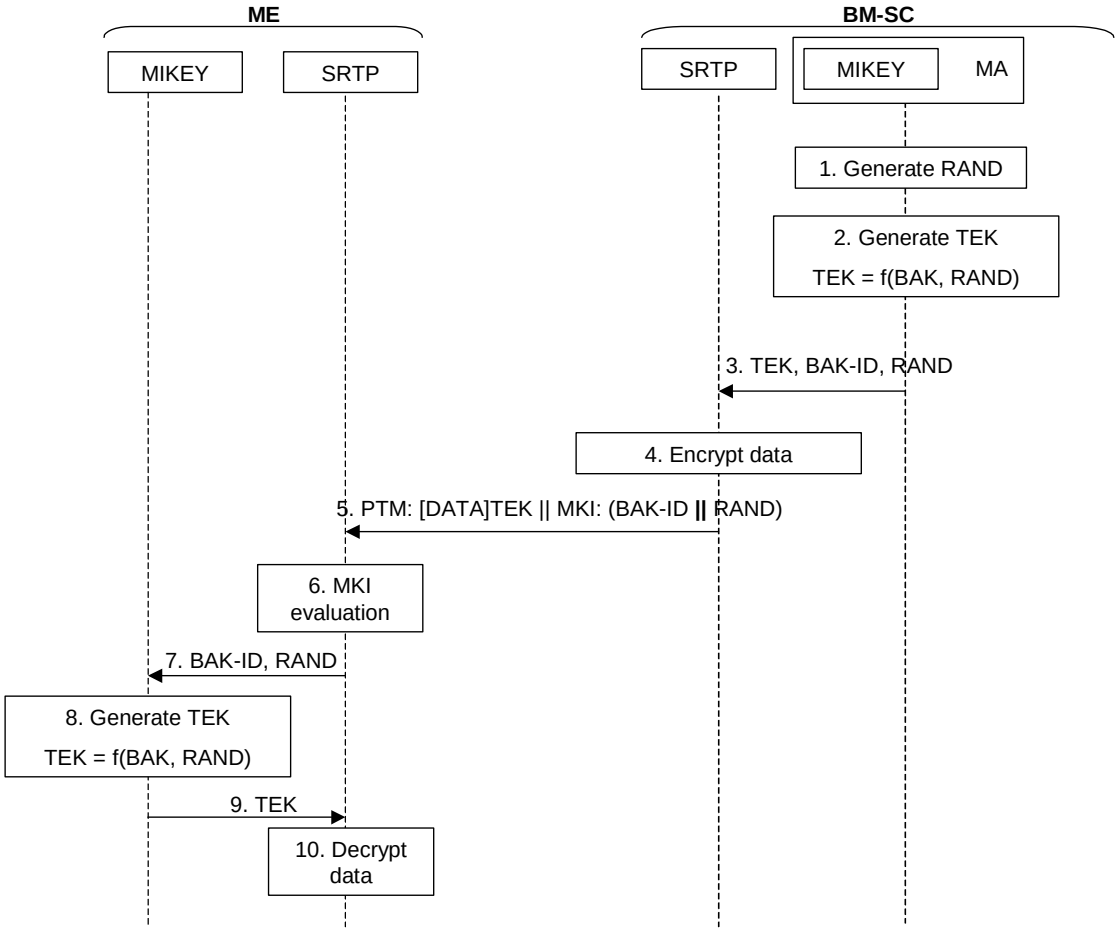

Figure 4: PTM re-keying and data transmission procedure

The initial keying is assumed to have happened.

1. The BM-SC generates RAND. Note that this can be generated frequently based on operator policy, e.g. every ten minutes.
2. MIKEY in BM-SC generates TEK using current BAK and RAND.
3. MIKEY passes TEK with BAK-ID and RAND to the security protocol (i.e. SRTP). Note that the concatenation of BAK-ID and RAND constitute the MKI field that is carried in SRTP header.
4. SRTP encrypts MBMS data with TEK. SRTP also includes the MKI field, i.e. the concatenation of BAK-ID and RAND to the packet.
5. SRTP sends the encrypted MBMS data to the UE.
6. SRTP in ME receives the encrypted content and takes the following actions:
   - If the MKI field is unchanged from the previous received content, processing continues in step 10.
   - If the MKI field is changed, processing continues in step 7.
7. SRTP passes the MKI to MIKEY IN ME and requests for a new TEK. Note that if the BAK-ID is not found in MIKEY's database, it may trigger re-keying procedure towards the BM-SC, see chapter 6.2.4.
8. MIKEY in ME calculates a new TEK using BAK and RAND.

9. MIKEY in ME passes the new TEK to SRTP.

10. SRTP decrypts the content using the TEK assigned for the content and passes the unencrypted content to the user application.

6.2.4 Point to Point re-keying procedure

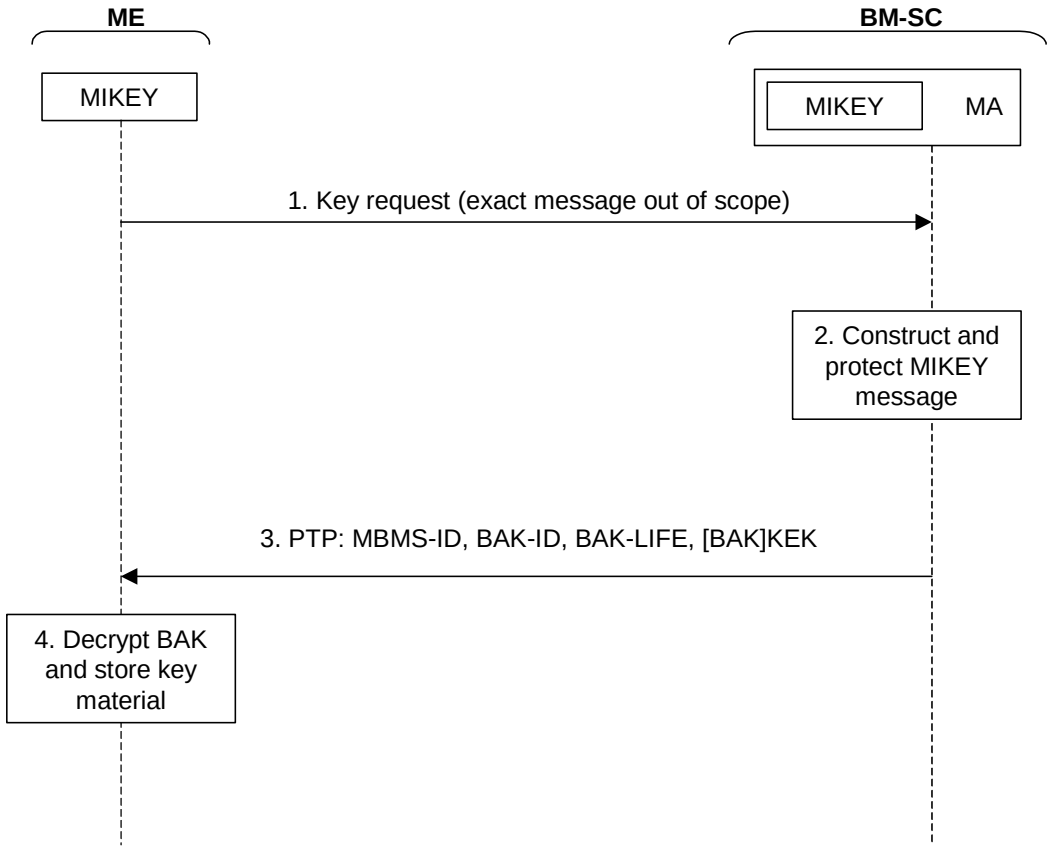

Figure 5: PTP re-keying procedure

1. ME detects the need for a new BAK for example from BAK lifetime or from MKI field when a BAK corresponding to a BAK-ID is not available. ME sends a re-key request to MA in BM-SC.
Note that the exact message is out of the scope of key management and it may be e.g. an HTTP request. The parameters needed are also FFS.

2. The MA in BM-SC receives the request. It is assumed that in this point the request message has been authenticated and authorised.

MIKEY message is constructed in BM-SC: the MBMS service ID that identifies the service is included into CSB field in MIKEY header, the BAK is encrypted with KEK, the BAK lifetime (BAK-LIFE) and BAK identifier (BAK-ID) are included as general extension payload field in MIKEY message.
Note that MIKEY has MAC checksum that covers the whole MIKEY message and a timestamp to protect against replay attacks.

The MIKEY message is as follows [MIKEY]:

MIKEY message = HDR, T, RAND, [IDi], {SP}, GEP (Key-id-1, Life-1), KEMAC (Key-id-1, Key-1),

where GEP is general extension payload, which can carry the key-id and lifetime for a key. GEP are added before the KEMAC since KEMAC includes the MAC for the whole MIKEY message. The Key-id-x inside the KEMAC is optional.

3. BM-SC sends the MIKEY message to the UE is point-to-point manner.

4. UE receives the MIKEY message and verifies the timestamp and MAC checksum. If they are correct, the message is processed. UE decrypts the BAK with KEK and stores the key material.

6.2.5 BAK lifetime

The TGK lifetime is defined by operator policy. Frequent TGK updates provide more security and allow more flexible subscription management, but can also cause more signalling overhead since UEs retrieve new TGK values in point-to-point manner.

6.2.6 Sending a group of keys in one message

When several keys are sent in one message, general extension payloads are used to convey the key-ids and lifetimes outside the encrypted part, the message is as follows:

MIKEY message = HDR, T, RAND, [IDi], {SP}, GEP (Key-id-1, Life-1), GEP (Key-id-2, Life-2), KEMAC (Key-id-1, Key-1, Key-id-2, Key-2),

where GEP is general extension payload, which can carry the key-id and lifetime for a key. GEPs are added before the KEMAC since it includes the MAC for the whole MIKEY message. The Key-id-x inside the KEMAC is optional in MIKEY.

6.2.7 Updating the BAK before use

A new BAK should be provided to ME before the new BAK value is needed to derive new TEK values. The UE retrieves the new BAK from the BM-SC. If many UEs try to retrieve the key simultaneously, there will be a burst of requests. This is so called implosion problem, which may be mitigated for example in the following ways:

7 Many TGKs may be sent in one MIKEY message

8 The UEs may be scheduled to retrieve the BAK at different times, e.g.:

- UEs may determine a random point in time before the BAK lifetime expires
- BM-SC may determine the retrieve time or time interval
- A combination of the above

6.4 UICC based key management

6.4.1 Overview

The ME based MBMS key management architecture in depicted in figure 2.

- The SM is general session manager in BM-SC that uses key management services of MIKEY protocol.

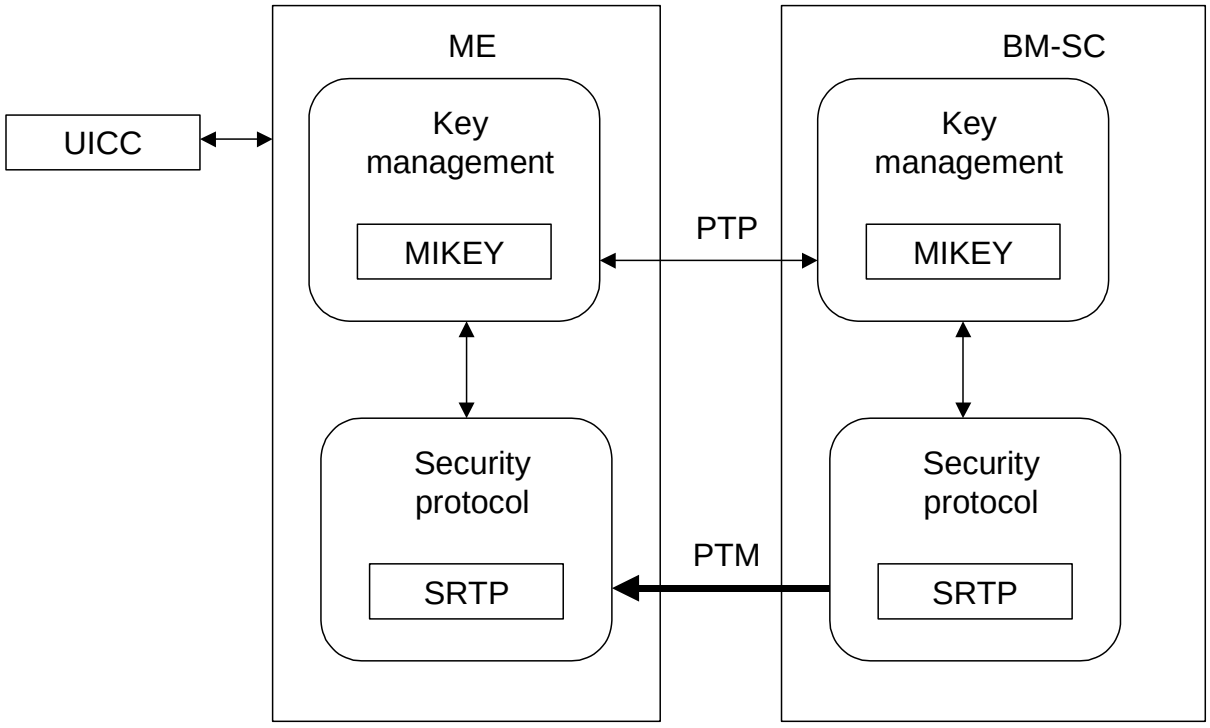

Figure 2: UICC based MBMS key management

6.4.2 BAK generation

The MA generates a value for Broadcast Access Key (BAK) and a value for RAND. These actions are independent of the actions of individual UEs.

6.4.3 Point to Point initial keying procedure

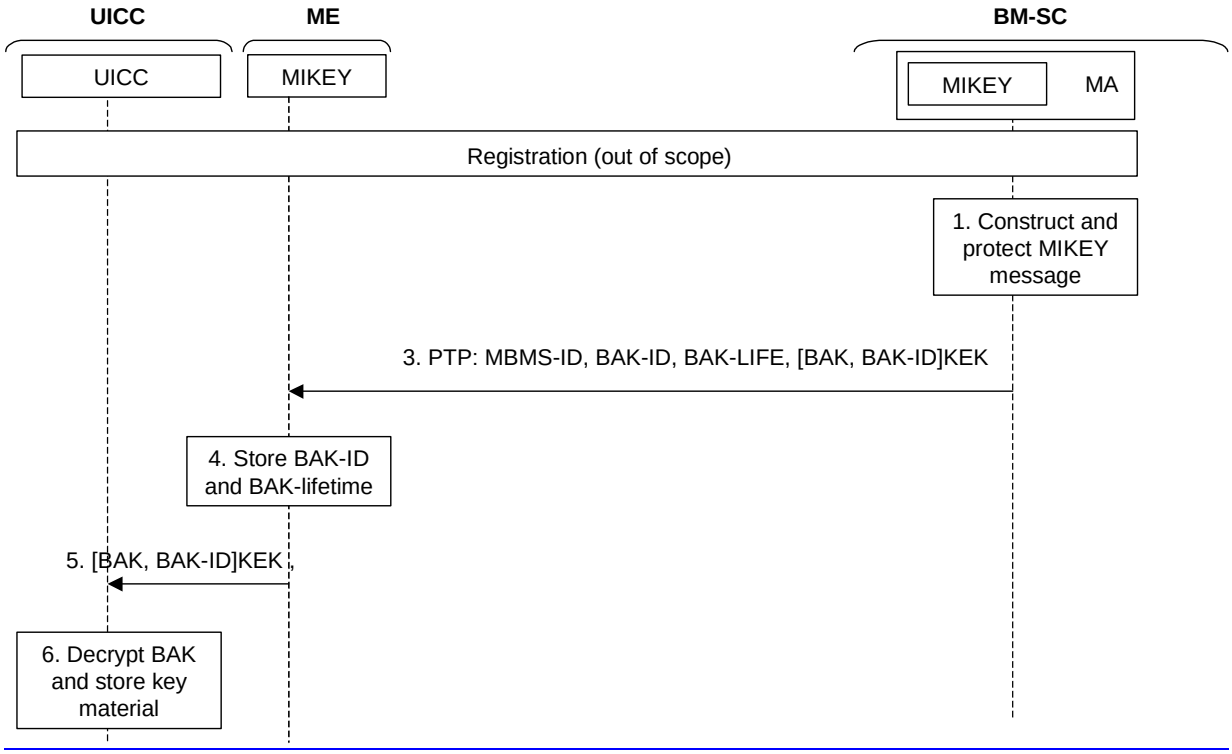

Figure 3: Initial keying procedure

Before the initial key exchange is performed, the user has been registered to the service and he/she has been authenticated and authorised for the requested MBMS service.

Editor's note: The authentication of the user is FFS and it could be done e.g. with GBA. Therefore it is out of the scope of current key management description.

Editor's note: The provision of KEK to UICC and BM-SC is FFS.

5. MA in BM-SC and the ME are provisioned with Key encryption key (KEK). The KEK is used to protect the MIKEY message delivery.

Editor's note: It is FFS how the KEK is generated, but it may be derived e.g. from the GBA keys. Also MIKEY can be used to generate the KEK from pre-shared keying material.

6. MIKEY message is constructed in BM-SC: the MBMS service ID that identifies the service is included into CSB field in MIKEY header, the BAK is encrypted with KEK, the BAK lifetime (BAK-LIFE) and BAK identifier (BAK-ID) are included as general extension payload field in MIKEY message.
Note that MIKEY has MAC checksum that covers the whole MIKEY message and a timestamp to protect against replay attacks.

The MIKEY message is as follows [MIKEY]:

MIKEY message = HDR, T, RAND, [IDi], {SP}, GEP (Key-id-1, Life-1), KEMAC (Key-id-1, Key-1),

where GEP is general extension payload, which can carry the key-id and lifetime for a key. GEP are added before the KEMAC since KEMAC includes the MAC for the whole MIKEY message. The Key-id-x inside the KEMAC is optional.

7. BM-SC sends the MIKEY message to the UE is point-to-point manner.

8. ME receives the MIKEY message and verifies the timestamp and MAC checksum. If they are correct, the message is processed. ME stores BAK-ID and BAK-lifetime.

9. ME passes the encrypted key material (BAK and BAK-ID) to the UICC.

10. UICC decrypts the BAK with KEK and stores the key material.

6.4.4 PTM re-keying and data transmission procedure

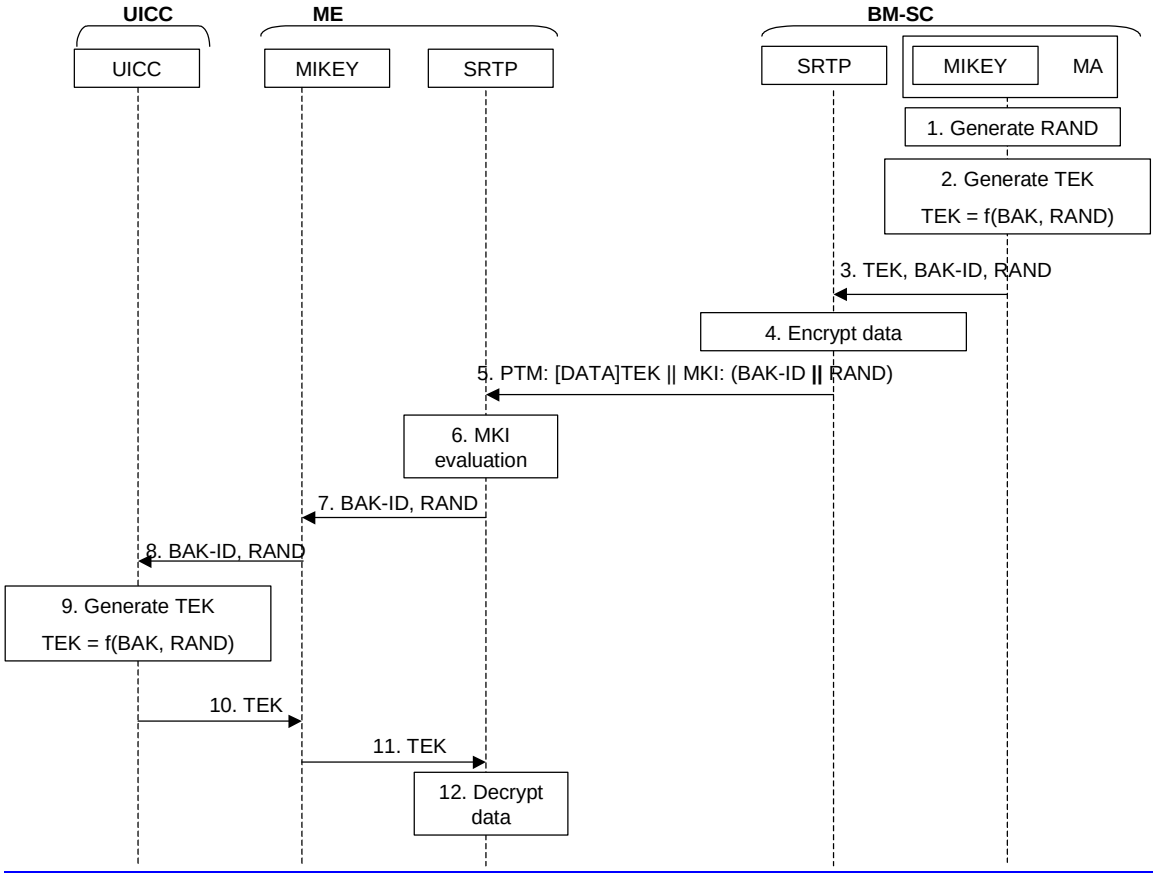

Figure 4: PTM re-keying and data transmission procedure

The initial keying is assumed to have happened.

11. The BM-SC generates RAND. Note that this can be generated frequently based on operator policy, e.g. every ten minutes.

12. MIKEY in BM-SC generates TEK using current BAK and RAND.

13. MIKEY passes TEK with BAK-ID and RAND to the security protocol (i.e. SRTP). Note that the concatenation of BAK-ID and RAND constitute the MKI field that is carried in SRTP header.

14. SRTP encrypts MBMS data with TEK. SRTP also includes the MKI field, i.e. the concatenation of BAK-ID and RAND to the packet.

15. SRTP sends the encrypted MBMS data to the UE.

16. SRTP in ME receives the encrypted content and takes the following actions:

- If the MKI field is unchanged from the previous received content, processing continues in step 10.
- If the MKI field is changed, processing continues in step 7.

17. SRTP passes the MKI to MIKEY IN ME and requests for a new TEK. Note that if the BAK-ID is not found in MIKEY's database, it may trigger re-keying procedure towards the BM-SC, see chapter 6.2.4.

18. ME passes the BAK-ID and RAND to the UICC.

19. UICC calculates a new TEK using BAK and RAND.

20. UICC passes the new TEK to ME.

21. ME passes the new TEK to SRTP

22. SRTP decrypts the content using the TEK assigned for the content and passes the unencrypted content to the user application.

6.4.5 Point to Point re-keying procedure

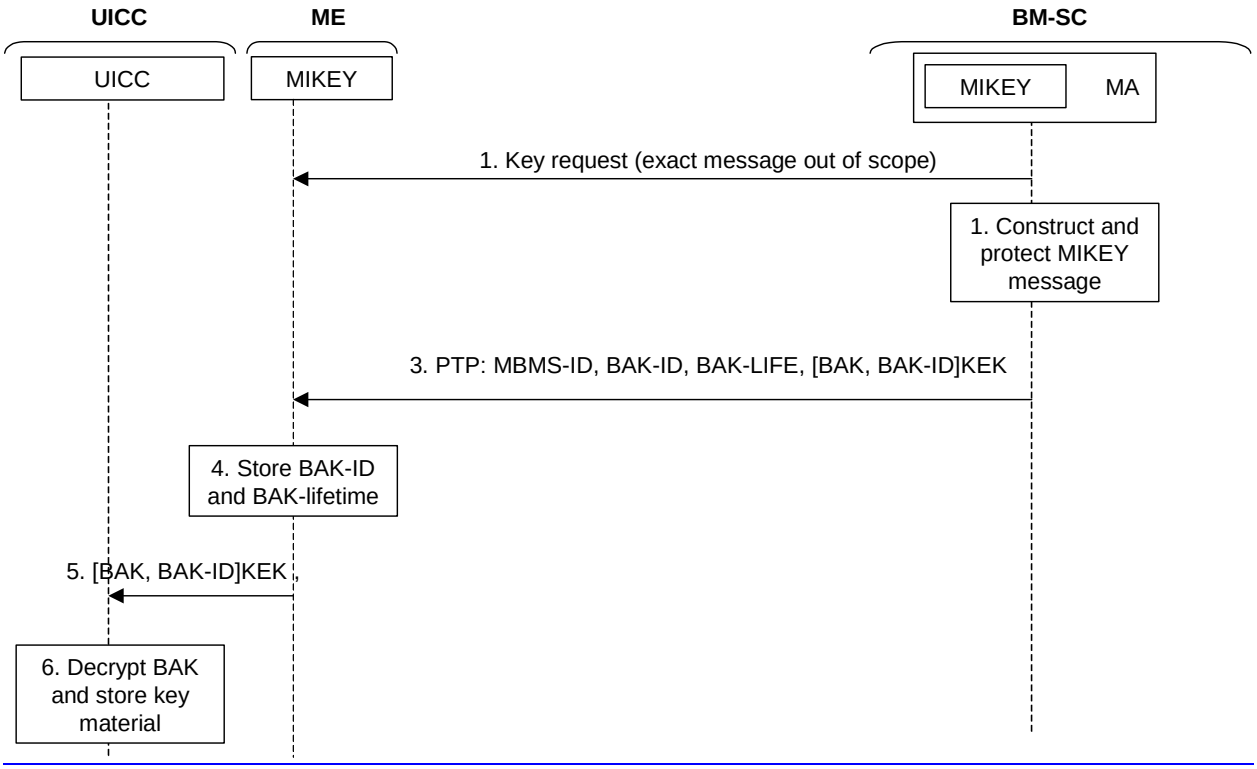

Figure 5: PTP re-keying procedure

5. ME detects the need for a new BAK for example from BAK lifetime or from MKI field when a BAK corresponding to a BAK-ID is not available. ME sends a re-key request to MA in BM-SC.
Note that the exact message is out of the scope of key management and it may be e.g. an HTTP request. The parameters needed are also FFS.

6. The MA in BM-SC receives the request. It is assumed that in this point the request message has been authenticated and authorised.

MIKEY message is constructed in BM-SC: the MBMS service ID that identifies the service is included into CSB field in MIKEY header, the BAK is encrypted with KEK, the BAK lifetime (BAK-LIFE) and BAK identifier (BAK-ID) are included as general extension payload field in MIKEY message.
Note that MIKEY has MAC checksum that covers the whole MIKEY message and a timestamp to protect against replay attacks.

The MIKEY message is as follows [MIKEY]:

MIKEY message = HDR, T, RAND, [IDi], {SP}, GEP (Key-id-1, Life-1), KEMAC (Key-id-1, Key-1),

where GEP is general extension payload, which can carry the key-id and lifetime for a key. GEP are added before the KEMAC since KEMAC includes the MAC for the whole MIKEY message. The Key-id-x inside the KEMAC is optional.

7. BM-SC sends the MIKEY message to the UE is point-to-point manner.

8. ME receives the MIKEY message and verifies the timestamp and MAC checksum. If they are correct, the message is processed. ME stores BAK-ID and BAK-lifetime.

9. ME passes the encrypted key material (BAK and BAK-ID) to the UICC.

10. UICC decrypts the BAK with KEK and stores the key material.

6.4.6 BAK lifetime

The TGK lifetime is defined by operator policy. Frequent TGK updates provide more security and allow more flexible subscription management, but can also cause more signalling overhead since UEs retrieve new TGK values in point-to-point manner.

6.4.7 Sending a group of keys in one message

When several keys are sent in one message, general extension payloads are used to convey the key-ids and lifetimes outside the encrypted part, the message is as follows:

MIKEY message = HDR, T, RAND, [IDi], {SP}, GEP (Key-id-1, Life-1), GEP (Key-id-2, Life-2), KEMAC (Key-id-1, Key-1, Key-id-2, Key-2),

where GEP is general extension payload, which can carry the key-id and lifetime for a key. GEPs are added before the KEMAC since it includes the MAC for the whole MIKEY message. The Key-id-x inside the KEMAC is optional in MIKEY.

6.4.8 Updating the BAK before use

A new BAK should be provided to ME before the new BAK value is needed to derive new TEK values. The UE retrieves the new BAK from the BM-SC. If many UEs try to retrieve the key simultaneously, there will be a burst of requests. This is so called implosion problem, which may be mitigated for example in the following ways:

9 Many TGKs may be sent in one MIKEY message

10 The UEs may be scheduled to retrieve the BAK at different times, e.g.:

- UEs may determine a random point in time before the BAK lifetime expires
- BM-SC may determine the retrieve time or time interval
- A combination of the above

3GPP TSG-SA3 Meeting #32
Edinburgh, Scotland, 9-13 February 2004

Tdoc ⌘S3-04xxxx

CR-Form-v7

CHANGE REQUEST

⌘ **TS 33.246** **CR** **CRNum** ⌘**rev** **-** ⌘ Current version: 1.0.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [X]

Title:	⌘	Support of key lifetime and batch of keys
Source:	⌘	Ericsson
Work item code:	⌘	MBMS
Date:	⌘	30/01/2004
Category:	⌘	**C**
Release:	⌘	Rel 6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)

Reason for change:	⌘	An important parameter in avoiding key request implosion problem is the key lifetime. The UEs can for example spread the key requests randomly over the lifetime of the key. If the UEs have no knowledge of the key lifetime, they will most probably request the key when it changes in the multicast data, thereby causing a peak of key requests. It is also seen beneficial to be able to send many keys to the UE in one batch. This can for example decrease the key signalling load, decrease the risk for peaks in key requests and also enable various charging schemes.
Summary of change:	⌘	Add a requirement on key lifetime to TS 33.246. Add a requirement on possibility of sending a batch of keys to TS 33.246.
Consequences if not approved:	⌘	Congestion in the system when users request new key simultaneously.

Clauses affected:	⌘	4.1.4

		Y	N		
Other specs Affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

4.1.4 Requirements on MBMS Key Management

R5a: The transfer of the MBMS keys between the MBMS key generator and the UE shall be confidentiality protected.

R5b: The transfer of the MBMS keys between the MBMS key generator and the UE may be integrity protected.

R5c: The UE and MBMS key generator shall support the operator to perform re-keying as frequently as it believes necessary to ensure that:

- users that have joined a multicast service, but then left, shall not gain further access to the multicast service without being charged appropriately;
- users joining a multicast service shall not gain access to data from previous transmissions in the multicast service without having been charged appropriately;
- the effect of subscribed users distributing decryption keys to non-subscribed users shall be controllable.

R5d: Only authorized users that have joined an MBMS multicast service shall be able to receive MBMS keys delivered from the MBMS key generator.

R5e: The MBMS keys shall not allow the BM-SC to infer any information about used UE-keys at radio level (i.e. if they would be derived from it).

R5f: All keys used for the MBMS service shall be uniquely identifiable. The identity may be used by the UE to retrieve the actual key (based on identity match, and mismatch recognition) when an update was missed or was erroneous/incomplete.

Editor's note: If ptm re- keying is used, the keys shall be delivered in a reliable way. Ptp re-keying is assumed to be reliable.

R5g: The BM-SC shall be aware of where all MBMS specific keys are stored in the UE (i.e. ME or UICC).

R5h: A UICC, realizing the function of providing session keys for decrypting the streaming data at the UE, shall only give session keys back to the UE if the input values used for obtaining the session keys were fresh (have not been replayed) and came from a trusted source.

R5i: There shall be a lifetime value associated to a high MBMS level key in order to support mechanisms that prevent all UEs from requesting the new high- level key simultaneously. This key lifetime shall be communicated to the UE with the associated key.

R5j: It shall be possible for the keying mechanism to send many keys (a batch of keys) with one keying message to the UE in order to decrease signalling load in the network and to support load balancing in key requests and different possible charging schemes.