3GPP TSG SA WG3 Security — S3#34 **S3-040554**

July 6 - 9, 2004

Acapulco, Mexico

Title: **Push and pull key management**

Source: **Ericsson**

Document for: **Discussion and decision**

Agenda Item:

Work Item: **MBMS**

1. Introduction

SA3 decided in meeting #33 that MIKEY [1] with extensions is used for MBMS key management. SA3 did not however decide on how pull/push will be used for MBMS key management. This contribution investigates push and pull issues with MIKEY. As was described in [2] MIKEY will be used both for MSK and MTK transport. The MSK transport is done in point to point manner while MTK transport is done in multicast manner multiplexed in the multicast data.

2. MTK transport

The MIKEY packets transporting MTKs are sent in multicast manner to the UEs. In multicast there typically is no feedback channel, thus push over UDP is a natural way to transport the MTK messages multiplexed in the multicast data.[1] The acknowledgement message of MIKEY should not be required for MTK messages since it would require a point to point feedback channel and could result in uplink congestion. Instead, MTK messages should be re-sent reasonably often in multicast to ensure that UEs receive the MTK. For the same reasons as above, if a UE detects that an MTK is missing, the UE should not request for MTK from the BM-SC, but should instead wait for the next MTK delivery in multicast channel.

3. MSK transport

Three alternatives to transport MSK are described and analysed below.

3.1 Push

It should be noted that MIKEY is designed originally for push key management, so it does not necessarily require a pull message from the UE. The push MSK management may be implemented as follows.

1. The BM-SC pushes MIKEY message to the UE over UDP.
2. If the BM-SC has requested it, the UE may send a MIKEY acknowledgement message over UDP.

[1] Carrying MIKEY over UDP requires that a UDP port is defined for MIKEY in IETF.

3.2 Pull

The pull MSK management may be implemented as follows. The request message may be authenticated with a key derived from GBA keys. Note that the different transport alternatives for pull are analysed in 4.2.

1. The UE requests MSK from the BM-SC.
2. The BM-SC sends the MIKEY message to the UE.
3. If the BM-SC has requested it, the UE may send a MIKEY acknowledgement message

The first flow does not include a MIKEY message since MIKEY does not have a request message.

3.3 Push solicited pull

A special case of pull is a push solicited pull. In this case the BM-SC solicits the UE to pull the key with a certain key availability message. The push initiated pull MSK management may be implemented as follows. (Note messages 2, 3, and 4 are the same as messages 1, 2 and 3 in 3.2 above):

1. The BM-SC solicits the UE to request for a key.
2. The UE requests MSK from the BM-SC.
3. The BM-SC sends the MIKEY message to the UE.
4. If the BM-SC has requested it, the UE may send a MIKEY acknowledgement message

4. Discussion of MSK transport

4.1 Why stand-alone pull is not sufficient?

The pull may also be a stand-alone solution without push. However, a stand-alone pull solution has some disadvantages: Firstly, the well-known key request congestion problem needs to be solved, i.e. it has to be ensured that UEs do not request the next key at the same time. Secondly, the network has no means to force a re-keying for the UE(s) if that is needed for some reason, e.g. due to key compromise that would require re-keying of all users. Push-based scheme does not suffer from these disadvantages. On the other hand it has been widely acknowledged in SA3 that it is inevitable to support pull in MBMS, e.g. due to cases when the UE detects in the multicast traffic that is has not got the current key. Therefore a combination of push and pull seems to be an appropriate solution for MBMS key management.

4.2 MIKEY over HTTP or UDP when using pull?

A combination of push and pull seems to be needed in MBMS. In this section it t is assumed that UDP is used for push cases since HTTP does not define a push message. Different pull cases over HTTP or UDP are analyzed in subchapters below

4.2.1 A) MIKEY procedure over UDP

1. The UE requests MSK from the BM-SC with HTTP POST message
2. The BM-SC sends the MIKEY message over UDP to the UE
3. If the BM-SC has requested it, the UE may send a MIKEY acknowledgement message over UDP

UE requests the key from the BM-SC with HTTP POST message, which can be authenticated with Digest headers. In the following MIKEY procedure both MIKEY messages are carried over UDP. This can be seen as a special case of 3.1, i.e. it is a “pull solicited push”.

Pros:

- The MIKEY procedure would then be over the same transport (UDP) in push and pull cases

Cons:

- The request message HTTP/TCP and MIKEY/UDP procedure would use different transport protocol.
- The HTTP Digest authentication procedure is violated if the BM-SC does not send the HTTP 200 OK message to the UE
- The BM-SC is not able to send the next nonce value of HTTP Digest to the UE. This could be useful when the UE tries to contact the BM-SC next time to avoid re-authentication. If the UE does not get the next nonce value, the BM-SC will likely challenge the UE next time it contacts the BM-SC and send HTTP 401 Unauthorized message. This will add one round trip to the procedure.

4.2.2 B) MIKEY procedure over HTTP

1. The UE requests MSK from the BM-SC with HTTP POST message.
2. The BM-SC sends the MIKEY message over HTTP 200 OK to the UE
3. If the BM-SC has requested it, the UE may send a MIKEY acknowledgement message over HTTP POST

UE requests the key from the BM-SC with HTTP POST message, which can be authenticated with Digest headers. The MIKEY procedure is sent over HTTP.

Pros:

- Transport would be the same within the pull procedure, i.e. for HTTP request and MIKEY procedure
- HTTP digest procedure is not violated
- HTTP provides more reliable transport

Cons:

- MIKEY is transported over different protocols for push and pull cases
- BM-SC is not able to send the next nonce value to the UE. This could be useful when the UE tries to contact the BM-SC next time to avoid re-authentication.
- Using HTTP means more overhead than UDP, but this is not considered an issue since HTTP is used only in (rare) special cases.

4.2.3 C) MIKEY procedure over HTTP and UDP

1. The UE requests MSK from the BM-SC with HTTP POST message
2. The BM-SC sends the MIKEY message over HTTP 200 OK to the UE
3. If the BM-SC has requested it, the UE may send a MIKEY acknowledgement message over UDP

UE requests the key from the BM-SC with HTTP POST message, which can be authenticated with Digest headers. The MIKEY key delivery message is sent over HTTP and acknowledgement is sent over UDP.

Pros:

- The MIKEY acknowledgement would come to the same place as in push case, i.e. to the UDP port. (This would require a priori agreed UDP port also for the acknowledgement since there is no source port due to the fact that key delivery MIKEY message was sent over HTTP.)
- HTTP digest procedure is not violated
- BM-SC is able to send the next nonce value to the UE. This may be useful when the UE tries to contact the BM-SC next time to avoid re-authentication.

Cons:

- The transport protocol would change from HTTP to UDP in the middle of MIKEY procedure. This means that the UE would receive the MIKEY message over HTTP/TCP but the acknowledgement would be sent over UDP.
- The MIKEY acknowledgement has to be sent to a beforehand agreed UDP port the since there is no source port due to the fact that key delivery MIKEY message was sent over HTTP.

4.2.4 D) Separate HTTP and MIKEY procedures

1. The UE requests MSK from the BM-SC with HTTP POST message
2. The BM-SC replies with HTTP 200 OK message
3. The BM-SC sends the MIKEY message over UDP to the UE
4. If the BM-SC has requested it, the UE may send a MIKEY acknowledgement message over UDP

UE requests the key from the BM-SC with HTTP POST message, which can be authenticated with Digest headers. The BM-SC replies with HTTP 200 OK message. In the following MIKEY procedure both MIKEY messages are carried over UDP. This can be also seen as a special case of 3.1, i.e. it is a “pull solicited push”.

Pros:

- The HTTP Digest authentication procedure is not violated as in 4.2.1
- The MIKEY procedure would then be over the same transport (UDP) in push and pull cases
- The BM-SC is able to send the next nonce value to the UE. This may be useful when the UE tries to contact the BM-SC next time to avoid re-authentication

Cons:

- The request message HTTP/TCP procedure and MIKEY/UDP procedure would use different transport protocol.

4.2.5 Analysis

Alternative A) cannot be considered since it violates the HTTP digest procedure. In D) the MIKEY messages are not carried in HTTP messages, which lead to disadvantages in B) and C). Instead in D) the HTTP request/ response and MIKEY procedures are separate which seems to provide a clean solution that does not suffer from the drawbacks of the other alternatives. Alternative D) has one more message than B) or C) since HTTP 200 OK and MIKEY are sent separately, but this overhead is considered insignificant.

4.3 Need for push solicited pull

The push initiated pull method, although requires more roundtrips and thus more bandwidth, offers some functional advantages compared to the strict push method:

1. Updating the MUK during the service
   The BM-SC may want to update the MUK or authenticate the UE during the MBMS service. Currently push method does not allow this. With push solicited pull the BM-SC could solicit the UE to authenticate itself.
2. Missing acknowledgement
   A problem might occur in push method that a malicious UE does not acknowledge the key delivery. Can it then be charged for the key? Push initiated pull could offer a solution to this, since BM-SC informs the UE of available key and UE requests it. Thus if the UE does not acknowledge the key and it does not request it again, it could be considered implicitly acknowledged.

The drawback of this method is that there is no message defined for the solicit purpose. One possibility could be to use the MIKEY message to this purpose. For example, a special value could be reserved in the key-id in the extension header to indicate to the UE that it should request for a new MSK and therefore also authenticate itself to the BM-SC.

5. Conclusion and proposal

This contribution has analyzed how to use push and pull for MTK and MSK transport.

For MTK transport in multicast, it is proposed that:

- MIKEY is carried over UDP
- MIKEY acknowledgement messages are not allowed for MTK transport
- UEs will not request for missing MTKs from the BM-SC, but will wait for the next MTK delivery

MSK key management seems to require a combination of push and pull.

For MSK transport in point to point, it is proposed that:

- MIKEY over UDP is used for normal MSK updates
- In special cases, e.g. when retrieving the initial MSKs or missing MSKs, the UE requests for the new MSKs with HTTP POST (with Digest headers). The HTTP procedure is followed by pushing MIKEY over UDP to the UE.
- A special value, e.g. 0x0, is reserved from the key identifiers in the MIKEY extension header to indicate to the UE that it should request for the key and thus authenticate itself.

The changes are implemented in the attached pseudo CR.

6. References

[1] MIKEY, internet draft, draft-ietf-msec-mikey-08.txt

[2] TD S3-040258, Extension payloads to MIKEY to support MBMS, Ericsson, SA3#33