3GPP TSG SA WG3 Security — S3#29 **S3-030356**

15-18 July 2003

San Francisco, USA

Agenda item: MBMS Security and 3GPP-3GPP2 joint meeting

Source: Qualcomm Europe

Title: MBMS Security Framework

Document for: Discussion

1 Introduction

3GPP2 has defined an efficient and flexible security architecture for Multicast-Broadcast services, based on a two-tiered hierarchy of symmetric keys: a long-term broadcast access key BAK delivered to the UICCs of subscribers to particular services, from which short-term keys SK are derived to encrypt traffic. This SK may be updated as frequently as desired, via broadcast.

This document describes the 3GPP MBMS Security Framework as it may be adopted from the 3GPP2 Broadcast-Multicast Security Framework, based on document *3GPP2 S.P0083 Version 0.5.*

2 Definitions and Abbreviations

Broadcast Access Key: A 128-bit key that provides access to the content of a particular multicast service/session for a certain amount of time (for example, one day, week or month). Each encrypted multicast service should have a different BAK value. Each BAK should have the associated BAK_ID, BAK_Expire:

Broadcast Access Key Identifier: A sequence number that identifies which value of BAK is currently being used to generate SK values for a particular multicast service. For a particular BAK, the corresponding value of BAK_ID is the same for all users.

Broadcast Access Key Expire: Indicates when the BAK will expire. This may be indicated by the time when the BAKwill expire, or by the Delta time left until the BAK expires, in which case it is calculated when a new BAK is received. Therefore, there may be multiple BAK_Expire values associated with the same BAK. The UE is expected to request a new BAK value before the BAK_Expire time expires.

Broadcast Access Key Distributor: (functional entity): Obtains authorization and TK from SM and encrypts BAK for delivery to UICCs.

Broadcast Access Key Generator: (functional entity): Generates BAK, determines BAK_ID and BAK_Expire, and delivers them to the BAKD and SKM. BAKs should be generated in a secure manner and appear random.

Content Encryptor: (functional entity): Encrypts content, and sends the encrypted content to the UE via the cellular system.

Content Provider: A third party content provider.

Registration Key: provisioned in the UICC and SM prior to joining a multicast service. Each SM should have a unique 128-bit RK for each UICC.

Short-term Key: The CE to encrypt the multicast data using the 128-bit short-term key (SK). The UE decrypts the multicast data using SK. The SK is calculated in the SKM and the UICC. The SK is derived from SK_RAND and BAK.

Short-term Key Random Number: used to calculate a unique SK

Short-term Key Manager: (functional entity): Generates SK using BAK and SK_RAND and passes SK, SK_RAND and BAK_ID to the CE.

Subscription Manager: (functional entity): performs accounting, authentication and authorization for MBMS. The SM also calculates the TK, based on the RK, used to hide the BAK. The SM may be the subscriber's home AAA (H-AAA) or be an independent entity.

Temporary Key: A 128-bit key used by the BAKD to encrypt BAK when provisioning BAK in the UICC. The TK is obtained from the SM.

2.1 Abbreviations

BAK	Broadcast Access Key
BAK_ID	BAK Identifier
BAK_Expire:	Indicates when the BAK will expire.
BAKD	BAK Distributor (functional entity)
BAKG	BAK Generator (functional entity)
CE	Content encryption (functional entity)
CP	Content provider
MBMS	Multimedia Broadcast/Multicast Service
RK	Registration Key
SK	Short-term Key
SK_RAND	SK Random Number use to calculate a unique SK
SKM	SK Manager (functional entity)
SM	Subscription Manager (functional entity)
TK	Temporary Key

3 Overview of MBMS Security Framework

3.1 MBMS Key Management

For the purposes of this document, the User Equipment (UE) is considered as two separate entities: the UICC and the Mobile Equipment (ME). The UICC is a low power processor that contains secure memory. The ME contains a high power processor, but no secure memory. The ME and the UICC may be integrated within one physical unit.

The user authentication and authorization for cellular services, as well as network security is outside the scope of this document. This document assumes the following trust model. all network entities are presumed to trust the other network entities that they communicate with, all communication between network entities is presumed to be secure, and all network entities are presumed to perform their tasks correctly. The UICC is trusted by the SM and vice versa. All network entities trust the UICC to keep secrets and perform its tasks correctly. However, the UICC trusts only the SM.

The main threat addressed by this document is the threat of a user obtaining cheap and reliable access to the content of a particular content stream without being authorized for that particular content stream. To counter this threat, the content is delivered encrypted to the ME, and the decryption keys are provided only to those users who are authorized (subscribed) to receive MBMS. The primary focus of this document is the key management scheme for content streams that require authorization.

Figure 4.2. MBMS Key Management. Multiple SK values are generated for each BAK value. The SK values are generated by combining BAK with the SK_RAND value that is transmitted with the encrypted content.

The content is encrypted using a unique and frequently changing *Short-term Key* (SK). The ME decrypts the content using the same SK. The SK should be frequently changed to minimizing the impact of a “rogue shell” sharing its SK with unauthorized users. The SK is never transmitted over the air; it is derived by the UICC from a *Broadcast Access Key* (BAK) and a random value broadcasted along with the encrypted content.

In order for the user to decrypt the necessary BAK values, the user's UICC must share an RK with an SM. Provisioning of RK is outside the scope of this document. A temporary encryption key (TK) is derived from the RK by the SM, sent to the BAKD, and subsequently used by the BAKD to encrypt/decrypt the BAK.

The BAK is encrypted by the BAK Distributor (BAKD), which may be associated with the Content Source, with the Subscription Manager, or the visited cellular system. The same BAK is provided to all users subscribing to a particular content stream, and it is active for a pre-defined period of time. Once the UICC has obtained the encrypted BAK (from the BAKD), it recovers the BAK (with its own calculated TK), and then computes the SK value needed to decrypt the broadcast. The SK is then delivered to the ME for content decryption. The BAK is never divulged to the ME, and only the UICC with a correctly pre-provisioned RK can recover the BAK.

.

3.2 Security Functional Architecture

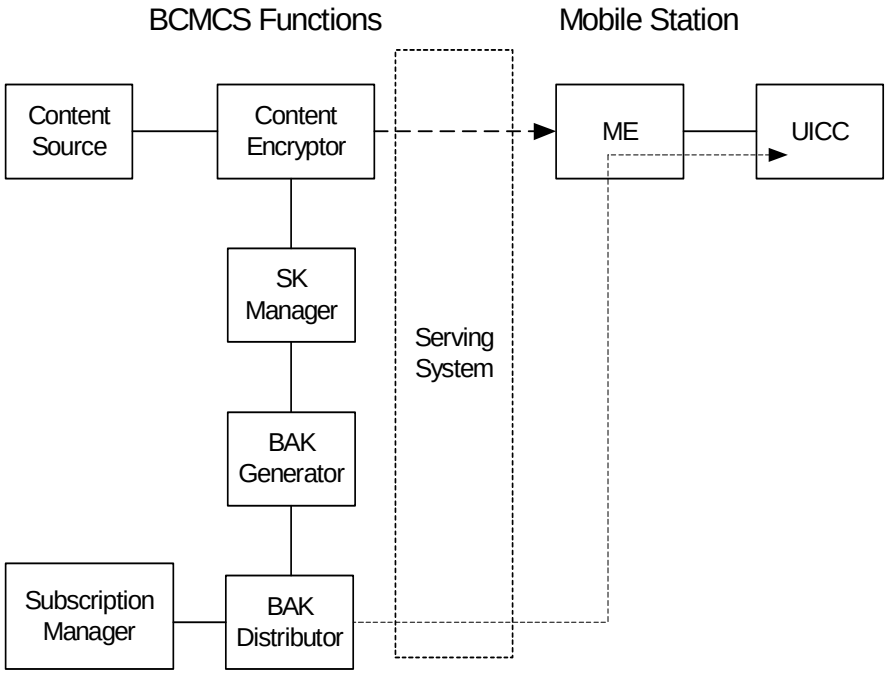

Figure 4.3 Functional Architecture

Figure 4.3 shows the functional entities that may be involved in MBMS. Those entities represent functions essential to support secure MBMS. They could be incorporated in one or more physical entities in the network. It should not be assumed that these functional entities correspond to separate architectural entities, so not all interfaces will require standardization. The allocation of the security functions to network entities is outside the scope of this document.

The MBMS functional entities are described in more detail below.

3.2.1 Summary of Functional Entities

CONTENT PATH

Content Source (CS): Generates the un-encrypted content.

Content Encryptor (CE): Encrypts the content using the SK provided by the SKM. Encryption may take place at the Content Source or in the serving cellular system.

KEY/AUTHORIZATION PATH

Subscription Manager (SM): May provide the functions of Authentication, Authorization and Accounting (AAA). The SM shares a Registration Key RK with the UICC: This key RK may be the A-key (which is the basis of key distribution and authentication for voice/data services as described in S.S0053), the key K used for AKA (as described in S.S0055), or some other key provisioned specifically for broadcast services. The SM calculates the TK, based on the user specific RK. *The provisioning of RK in the UICC and SM is beyond the scope of this document.*

BAK Generator (BAKG): Generates the Broadcast Access Key (BAK) as required, and distributes BAK and to BAKD and SKM.

BAK Distributor (BAKD): Controls the distribution of the Broadcast Access Key (BAK).

SK Manager (SKM): Controls the updating and distribution of the Short-term Key (SK).

User Equipment (UE) For the purposes of this document, the UE is considered as two separate entities, the UICC and ME.

UICC: User Identity Module : The UICC shares a key RK with the SM. The UICC performs all key management related to MBMS.

ME: Mobile Equipment. Includes equipment for receiving the broadcast. The ME performs decryption of encrypted content using SK obtained from the UICC.

3.2.2 Summary of Key Distribution

Figure 4.4 shows the basic communications involved in the key distribution for MBMS encryption. Many details are omitted in this diagram for the sake of clarity. The Figure is described below.

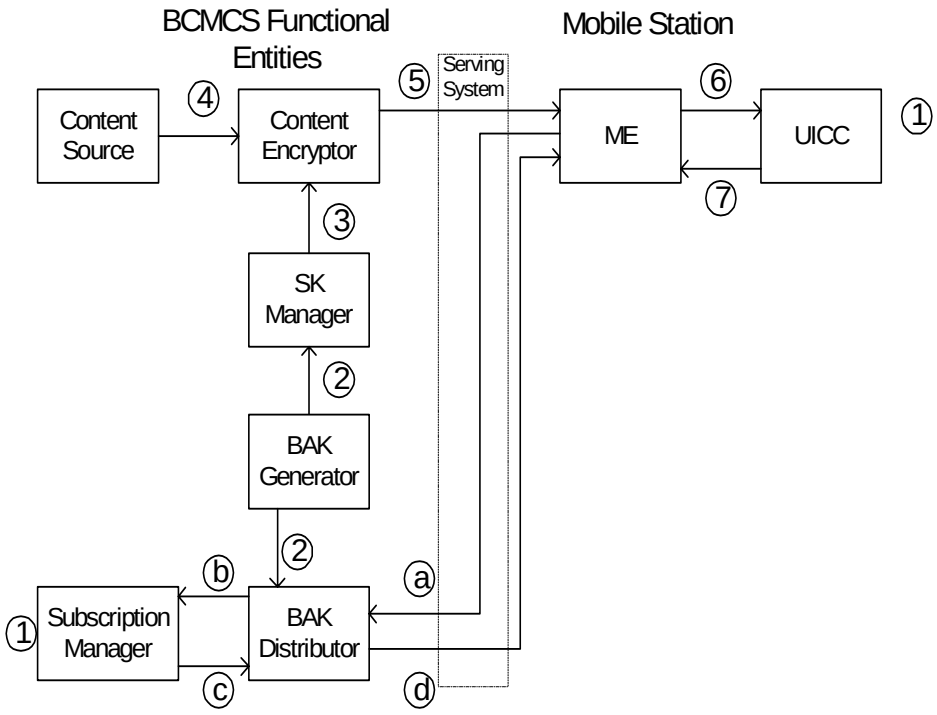

Figure 4.4 Logical Architecture showing functional entities involved in MBMS security.

3.2.2.1 BAK Generation

The following steps (1-2) are performed prior to the transmission of encrypted content to the UE:

1. The UICC and SM are provisioned with a Registration Key RK that will be the basis of authentication and key exchange with respect to MBMS.

2. The BAKG generates a value for BAK, and associates the value with an identifier BAK_ID and an expiry time (BAK_Expire). The value of BAK, along with the corresponding values of BAK_ID and BAK_Expire are passed to the SKM and BAKD.

3.2.2.2 Normal Procedure With Unchanged BAK

The following steps (3-7) occur under normal circumstances, when BAK is unchanged from the previous decryption operation in the UE.

3. The SKM creates SK from the current BAK and a random value SK_RAND. The SKM passes SK, SK_RAND, BAK_ID and BAK_Expire to the CE.

4. The CS sends unencrypted content to the CE.

5. The CE encrypts the content using SK and sends the encrypted content to the UE via the serving system. The CE also includes SK_RAND and BAK_ID with the encrypted content stream.

6. The ME receives the encrypted content, and takes action as follows:

   a. If BAK_ID and SK_RAND are unchanged from the last received content, the ME decrypts the content using the value of SK currently assigned to that content stream and passes the result to the user application;

   b. If BAK_ID or SK_RAND have changed, the ME requests a new SK from the UICC, including the broadcast service identifier, BAK_ID and SK_RAND.

7. The UICC generates SK from BAK and SK_RAND and returns SK to the ME, which decrypts the content and passes the result to the user application.

3.2.2.3 Procedure With Changed or Unavailable BAK

Steps (a-d) in Figure 4.3 are performed to request a new BAK. Section 4.4 discusses possible methods by which the UE determines that a new BAK is needed.

a. The ME sends a BAK request to the BAKD, including the BAK_ID of the BAK requested. The BAK request may include authentication information based on RK, which can be used by the SM to determine that the request came from a legitimate subscriber. Section 4.4.1 discusses the need for authentication of BAK requests.

b. In order to send BAK to the UICC, the BAK must be encrypted to protect it against reception by other than the intended recipient. The BAKD requests a temporary key (TK) generated by the SM.

c. The SM generates TK from a random value TK_RAND and RK. TK_RAND may be generated by the BAKD, or by the SM. TK_RAND may also be used as a challenge in the authentication process described in step a. The SM sends TK and TK_RAND to the BAKD.

d. The BAKD encrypts the value of BAK with TK, and sends the encrypted BAK, along with TK_RAND and BAK_LIFE to the UICC via the ME. The UICC first forms TK from TK_RAND and RK, and then decrypts the encrypted BAK with TK to form BAK. The value of BAK and its associated BAK_LIFE are stored in the UICC. The UICC should be able to store at least two values of BAK so that a new BAK can be obtained and stored in anticipation of the expiration of the BAK lifetime.

3.3 BAK Management

A user can repeat the BAK update process with multiple content streams. Thereafter (from a security perspective) the UE has the potential to be managing the decryption keys for multiple content streams simultaneously. In order to determine which BAK corresponds to which content stream, a MBMS identifier MBMS_ID should be stored with the BAK.

This document does not specify how an UE determines that it needs to update BAK. We assume that a means will be provided for the UE to determine that its BAK is about to expire or has expired, triggering

action to perform a BAK update. Several methods are possible for accomplishing this. The specific method used is not important for the present subject.

BAK_ID. The UICC can determine if a BAK is associated with a particular content stream by referring to the corresponding MBMS_ID. However, there will be times when the UICC needs to store two BAK values associated with a particular MBMS_ID. The reason is that the BAK must be updated in the UICC prior to the BAK actually being used to encrypt/generate SK values (see Section 4.4.3). Consequently, both the old BAK and the new BAK will reside in the UICC simultaneously. The UICC must have some means to determine which BAK is valid. The recommended method is to allocate a BAK identifier BAK_ID to each BAK, in addition to the MBMS_ID. the BAK is identified by the MBMS_ID and BAK_ID. The CE will include the BAK_ID with the encrypted content in the broadcast data. The UICC can now distinguish if the old BAK is still being used or if the new BAK has begun to be used.

A value for BAK_Expire will be provided along with the BAK, so the UE can update expired keys.

BAK lifetime. The BAKG decides how often BAK is changed. The following issues should be considered in deciding how often BAK is to be changed.

- Frequent BAK changes will provide more security.
- Frequent BAK changes will also provide greater flexibility in subscription control. We show this by example. Once a user has BAK, they can access the content for the lifetime of that BAK. Suppose the BAK is changed at the beginning of every month. If a user's subscription runs out halfway through the lifetime of a BAK, the user will still be able to generate SK (and thus view the content) until the BAK expires. So by changing BAK only every month, the SM can only charge subscriptions from the beginning of the month to the end of the month. A user can't subscribe from the middle of one month to the middle of the next. However, if BAK changed every day, then the user could subscribe from the beginning of any day during the month.
- Increasing the frequency of BAK changes should be evaluated against a possible increase in the number of times the mobile station has to retrieve new BAK values.

3.3.1 Authenticating BAK Requests

An adversary cannot obtain BAK by performing a BAK request while impersonating a subscribed user. Only the subscribed user will be able to derive TK from RAND_TK, and thus extract BAK. For this reason, the BAKD does not need to authenticate BAK requests in order to protect BAK.

If, however, other control functions or accounting data are included with the BAK request, then it is necessary to prove the authenticity of the data source. It may also be desirable to authenticate requests to prevent malicious generation of excess network traffic. *The procedures for authenticating the request are for further study.*

Note that even though authentication of BAK requests may not be needed, there is still a need for authorization of the request. Such authorization must be specific to the broadcast service to which the requested BAK applies.

3.3.2 Storage of BAK

It is essential to the security model used herein that the UICC does not reveal BAK. If a single UICC reveals BAK, then all security is compromised until the BAKG changes BAK.

The UICC should store BAK and related data about BAK, such as BAK_ID and expiration time, if any. The BAK is identified by the MBMS_ID (of the corresponding MBMS content stream) and BAK_ID.

The ME may store the BAK-related data, to save requesting this information from the UICC.

3.3.3 Updating BAK before use

It may prove beneficial to provision UICC with BAK shortly before BAK begins being used to derive SK values. Otherwise, once the CE starts sending packets with SK derived from the new BAK, the user would experience a delay as the UE performs a BAK update. If many users are tuned in, then there will be a burst of traffic as all the UEs perform a BAK update.

To avoid such problems, the BAKD should allow an UE to obtain the new BAK shortly before the BAK changes. Different UE may have different schedules for performing BAK updates, to prevent too many UEs performing a BAK update at once.

For security reasons, BAK should be distributed as close as possible to the time of use.

3.4 MBMS Security Algorithms

3.4.1 Encryption of Content

The specification of encryption algorithms is for further study.

3.4.2 Encryption of BAK

The specification of encryption of BAK (under key TK) is for further study.

3.4.3 Management of TK

3.4.3.1 TK_RAND

TK_RAND shall be 64 bits in length.

TK_RAND shall be generated in a manner that minimizes the probability that the same TK_RAND is used to create a TK for different BAK encryptions destined to the same terminal. The probability should be the same as for random selection of TK_RAND.

3.4.3.2 TK Generation

TK shall be 128 bits in length.

TK shall be generated from TK_RAND and RK using a secure one-way function chosen to minimize the likelihood that an adversary can obtain TK with knowledge of TK_RAND alone, and to minimize the likelihood that an adversary can obtain RK with knowledge of TK_RAND and TK. A standardized SHA-1 based function will be used for this purpose.

3.4.4 SK

3.4.4.1 SK_RAND

SK_RAND shall be 32 bits in length.

SK_RAND shall be generated in a manner that minimizes the probability that the same SK_RAND is used to create an SK for different content encryptions destined to the same terminal. The probability should be the same as for random selection of SK_RAND.

SK_RAND shall be generated in a manner that minimizes the probability that an adversary can guess the next value of SK_RAND with knowledge of previous values. This requirement makes it more difficult for a modified ME shell to compute future SK values and distribute the values to other MEs.

3.4.4.2 SK Generation

SK shall be 128 bits in length.

SK shall be generated from SK_RAND and BAK using a secure one-way function chosen to minimize the likelihood that an adversary can obtain SK with knowledge of SK_RAND alone, and to minimize the likelihood that an adversary can obtain BAK with knowledge of SK_RAND and SK.

4 Motivation

4.1 Design Philosophy

The major threat addressed in this document is that of user(s) accessing the broadcast content without authorization. This threat applies only when access is controlled on a subscription basis.

4.1.1 A Specific Goal

To access the broadcast content, a user must have the current decryption keys. The UICC is not powerful enough to decrypt the content so the ME must perform the decryption. This implies that the decryption keys must be stored in the ME, which cannot be considered a secure storage device. It must be considered that eventually an attacker may find a way to extract the current decryption key from the ME. An attacker who is a subscribed user will

then be able to distribute the decryption key to other non-subscribed users. In summary, the need to store decryption keys in unsecure memory makes it impossible to design a scheme where non-subscribed users CANNOT access the data.

We must recognize that the most we can do is dissuade the potential market (those users for which the service is targeted) from using illegitimate means to access the content.

Assuming the primary threat is subscribed users distributing decryption keys to non-subscribed users, the solution is for the decryption key to change frequently and in an unpredictable manner. The challenge is achieving this while minimizing the transmission overhead required for key distribution. The solution described herein is to take a two-tiered approach, distributing a Broadcast Access Key (BAK) to each user individually, and for many decryption keys to be derived using the BAK and public information sent with the broadcast. The BAK is stored in the UICC. This approach allows the short-term keys SK to be changed as frequently as desired, without incurring significant costs in radio resources or key management, while BAK is changed depending on business requirements stemming from the subscription periods for the services.